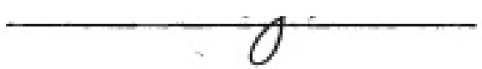
In presenting the dissertation as a partial fulfillment of the requirements for an advanced degree from the Georgia Institute of Technology, I agree that the Library of the Institution shall make it available for inspection and circulation in accordance with its regulations governing materials of this type. I agree that permission to copy from, or to publish from, this dissertation may be granted by the professor under whose direction it was written, or, in his absence, by the Dean of the Graduate Division when such copying or publication is solely for scholarly purposes and does not involve potential financial gain. It is understood that any copying from, or publication of, this dissertation which involves potential financial gain will not be allowed without written permission.



THE DETERMINATION OF EFFECTIVE DELAYED NEUTRON AND PHOTONEUTRON KINETICS PARAMETERS IN A HIGHLY ENRICHED HEAVY-WATER REACTOR

A THESIS

Presented to

The Faculty of the Graduate Division

by

Walter Waverly Graham, III

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy in the School of Nuclear Engineering

Georgia Institute of Technology

August, 1965

THE DETERMINATION OF EFFECTIVE DELAYED NEUTRON AND PHOTONEUTRON KINETICS PARAMETERS IN A HIGHLY ENRICHED HEAVY-WATER REACTOR

Approved:

Chairman

Date approved by Chairman: Aug 16, 1965

ACKNOWLEDGMENTS

It is a pleasure to acknowledge the assistance of at least some of the many people who have had an interest in the work described in this thesis. Dr. W. B. Harrison has provided generous support and, along with the other faculty members of the School of Nuclear Engineering, a considerable measure of encouragement. Dr. D. S. Harmer, my thesis advisor, and Dr. C. E. Cohn of the Argonne National Laboratory, my thesis consultant, have provided invaluable technical advice and direction. I am grateful to Dr. G. G. Eichholz and Dr. C. J. Roberts, who served as members of my reading committee, and who very kindly provided helpful suggestions, both during the course of the experimental work and the drafting of this manuscript. To Mr. R. E. Meek and his team of electronics technicians I am indebted for their diligence on my behalf. I would like to thank Mr. J. C. Gundlach of Reactor Controls, Inc. for his inspired response to my detector needs and Mr. F. W. Shue for genius in the machine shop of the Georgia Institute of Technology Nuclear Research Center. To my good friend, Mr. R. J. Johnson, I wish to express my sincere gratitude for hours of unselfish advice and assistance. He was also instrumental in the development of the methods described in Appendix B for assessing reactor power history. Mrs. Lydia Fisher has done an outstanding job of editing and typing this manuscript.

The research represented here has been supported by Atomic Energy Commission Special Fellowships in Nuclear Science and Engineering, administered by the Oak Ridge Institute of Nuclear Studies, by the Ford

Foundation Loan Fund, and by Atomic Energy Commission Research Contract No. AT(38-1)-403. I also acknowledge gratefully support by funds from the Georgia Tech School of Nuclear Engineering and the Nuclear Sciences Division of the Engineering Experiment Station.

Finally, to my wife, Ann, and to our children, Kerry and Holly, I want to express my appreciation for love and understanding when work took priority over conversation and story-books.

TABLE OF CONTENTS

TABLE OF CONTENTS (Concluded)

LIST OF TABLES

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

LIST OF ILLUSTRATIONS (Concluded)

SUMMARY

This research was undertaken in order to determine experimentally effective values of the delayed neutron and photoneutron relative abundances, $\beta_i \text{eff}/\Sigma_i \beta_i \text{eff}$, and the decay constants, λ_i, which are applicable to a highly enriched heavy-water nuclear reactor. These parameters are required for precise calculations of the kinetic behavior of the reactor. They have not previously been determined in a reactor of this specific type. Neutron flux decay data following a rod drop in the Georgia Tech Research Reactor were collected over periods in excess of three days. The reactor had not operated above 200 watts; most of its operation had been at 1 watt. Scintillation and fission chamber detection systems were used. The data have been fitted to the expression

$$n(t_j \rightarrow t_j + \Delta t_j) = \sum_{i=1}^{M} \int_{t_j}^{t_j+\Delta t_j} a_i' e^{-\lambda_i' t} \, dt$$

by the weighted least-squares criterion, using the Variable Metric Minimization method. Corrections have been made in order to account for delayed neutron precursor concentrations at the time of the rod drop, the finite time required to drop the rods, and for post-drop sub-critical neutron multiplication. Various values of M, the total number of groups permitted in the summation, have been tried, and the best fit was found for M = 13. One of the thirteen groups is attributed to background. Thus, twelve

delayed neutron groups are indicated for the Georgia Tech Research Reactor. Comparisons are made with previously obtained values of relative group abundances and decay constants available in the literature.

NOMENCLATURE

a_i fraction of count rate after the prompt drop attributable to the i^{th} delayed neutron group

a_i' count rate after the prompt drop attributable to the i^{th} delayed neutron group (counts/sec)

B total number of counting intervals

C_i concentration of the i^{th} delayed neutron group precursor (atoms/cm^3)

F sum of squares of weighted differences between collected and calculated counts for all intervals

k_{eff} effective neutron reproduction factor

k_p neutron reproduction factor component attributable to prompt neutrons

ℓ^* prompt neutron lifetime (sec)

M total number of delayed neutron groups

n reactor neutron concentration (neutrons/cm^3)

P reactor poisoning, the ratio of the number of thermal neutrons absorbed by the poison to those absorbed in the fuel

Q_i equilibrium fraction of the i^{th} delayed neutron precursor

R_i relative abundance of the i^{th} delayed neutron group

s_i i^{th} root of the inhour equation (sec^{-1})

t time (sec)

T instantaneous reactivity insertion time

T_D reactor flux doubling time (sec)

V weighted variance of fit, a statistical figure-of-merit for curve fitting which equals $\sum_{j=1}^{B} \chi_j^2/(B - 2M)$

y experimentally observed neutron count

NOMENCLATURE (Concluded)

α	general constant of exponential change (sec^{-1})
β	fraction of neutrons from a fission which are delayed
β_i	fraction of neutrons from a fission which are attributable to the i^{th} delayed neutron group
Δk	fractional change in the neutron reproduction factor
Δt	time interval
λ_i	decay constant of the i^{th} delayed neutron precursor (sec^{-1})
σ	standard deviation of data
Σ_a	macroscopic absorption cross section (cm^{-1})
ϕ	reactor neutron flux (neutrons/cm^2 sec)
χ	weighted difference between experimentally observed quantity and the calculated value for that quantity

CHAPTER I

INTRODUCTION

Background

Nuclear fission was discovered in 1939. In this process a fissionable isotope such as U-235 captures a neutron and within about 10^{-14} seconds[1] breaks up into two heavy fragments while emitting about 2 or 3 neutrons and about 5 gamma rays. The two heavy products of the fission process decay at characteristic rates until they reach stable isobars. The fraction of each isobar produced in fission is called the "yield." This yield varies from extremely small values up to about 0.06 per fission. The nuclide from which a given decay or isomeric transition product is produced is called that product's "precursor" or "parent."

The production of a self-sustaining chain reaction in 1942 depended upon the fissioning nucleus yielding a sufficient number of neutrons so that, on the average, one of those fission neutrons would go on to produce a subsequent fission. More than 99 percent of the neutrons originating in the fission process are emitted at the instant that the fission occurs. For this reason they are called "prompt" neutrons. But in the same year that fission was discovered, another category of fission-produced neutrons, "delayed" neutrons, was found by Roberts, Meyer and Wang.[2] Delayed neutrons are products of the fission event, but they appear after the moment of fission at a time that is long compared with the extremely short time scale over which prompt neutrons appear.

It has been generally concluded that the emission of delayed neutrons arises from the decay of excited fission-product nuclides, including some isotopes of bromine and iodine.[3] The process is evidently influenced by the shell structure of certain of these radioisotopes. For example, Kr-87, a known delayed-neutron emitter, contains 51 neutrons, one more than the number required to produce a closed "nuclear shell" structure. The binding energy of this extra neutron is 5.5 MeV.[4] The parent of Kr-87, Br-87, contains too many neutrons for stability and is a beta emitter. If the Kr-87 is formed with an excitation energy greater than this 5.5 MeV binding energy, it will emit the odd neutron and become Kr-86. Since this neutron is emitted as a result of a fission event which, on the average, occurred some seconds earlier, it is referred to as a "delayed neutron," distinguishing it from "prompt neutrons" which appear at the instant of fission. The beta decay of Br-87 does not always yield a daughter nucleus which is in the highly excited state required for delayed neutron emission. As a matter of fact, in at least 90 percent of the transitions,[4] Kr-87 decays by beta emission to Rb-87 which then decays to stable Sr-87 by another beta decay. If the highly excited state of the delayed neutron emitter, Kr-87, is obtained, however, then the neutron emission occurs immediately after the formation of the Kr-87, and the observed half-life of such a process in a reactor is that of the parent beta emitter.

In heavy-water- and beryllium-moderated reactors, another category of delayed neutrons may result from a fission event in a less direct manner. High-energy gamma rays may interact with deuterium or beryllium nuclei to emit neutrons. The threshold for such photoneutron production is 2.23 MeV for H-2 and 1.67 MeV for Be-9.[5] While photoneutron produc-

tion by prompt gamma rays occurs continuously in the operating reactor, these photoneutrons are indistinguishable from the prompt neutrons produced directly in the fission process. However, photoneutrons are particularly significant for kinetics considerations when they are produced by delayed gamma rays from fission products. The period of a delayed photoneutron (i.e. the mean time between a fission event and the subsequent absorption of the delayed neutron attributable to that fission event) is determined by precursor beta decay. Photoneutron periods are generally much longer than delayed-neutron periods.[6]

Delayed neutrons of both types are important to the control of a nuclear reactor. It has been mentioned that a self-sustaining chain reaction requires, on the average, that each fissioning nucleus produce enough neutrons so that one of these neutrons may itself cause a subsequent fission. That is, in order for the neutron population to remain constant, each neutron which is lost in a fission must be replaced by another which will cause a later fission. This replacement neutron may be either prompt or delayed. If the reactor is just critical or not-too-far super-critical, it requires the small fraction of delayed neutrons as replacements in order to maintain the neutron population, in the critical case, or to cause it to increase in the not-too-far super-critical case. Under these circumstances, the reactor is said to be "delayed critical," meaning that if the delayed neutron contribution were not present, the reactor would no longer be critical. When the super-critical reactor configuration is such that criticality does not depend on the presence of delayed neutrons, then the reactor is said to be "prompt critical."

The kinetic or time behavior of the reactor is determined by the

production rate and loss rate of neutrons and the time between neutron generations i.e. the time from the appearance of a neutron until the event in which it produces its successor. In spite of their small abundance, delayed neutrons control the time between neutron generations because of their long periods. There are different periods and thus half-lives associated with different "groups" of delayed neutrons. These depend upon the half-lives of the delayed neutron precursors. The significance of this effect is well illustrated in a numerical example presented by Glasstone and Edlund.[7]

Thus, since the role of delayed neutrons in setting the pace for reactor kinetic behavior is an established qualitative fact, it is appropriate that their role be examined in a quantitative manner.

Neutron population change in a reactor is frequently described by the space-independent zero-energy kinetics equations.[7] These equations describe the observable neutron density in a reactor before and after a change in control rod position or reactor configuration which affects the neutron reproduction factor, k_{eff}. The first equation,

$$\frac{dn}{dt} = \left[k_{eff}(1 - \beta) - 1 \right] \frac{n}{\ell^*} + \sum_{i=1}^{M} \lambda_i C_i \qquad (1)$$

describes the change in neutron level, n, as a function of time following a step insertion of reactivity. In this equation

n = reactor neutron concentration

t = time

k_{eff} = effective neutron reproduction factor

β = fraction of neutrons from a fission which are delayed

ℓ^* = prompt neutron lifetime

M = total number of delayed neutron groups

λ_i = decay constant of the i^{th} delayed neutron precursor

C_i = concentration of the i^{th} delayed neutron group precursor

Since a change in the number of fission events also changes the number of fission product nuclides produced, such a step change in reactivity also alters the supply of delayed neutrons. This can be expressed by describing the change in concentration of the neutron precursors as

$$\frac{dC_i}{dt} = \frac{k_{eff}\,\beta_i\,n}{\ell^*} - \lambda_i\,C_i \qquad (2)$$

where β_i is that fraction of neutrons from a fission which is attributable to the i^{th} delayed neutron group. The solution of these equations for neutron density as a function of time following a step insertion of negative reactivity is given in considerable detail by Cohn and Toppel[8] and results in the expression

$$n(t) = n(0) \sum_i a_i\, e^{-\lambda_i' t}$$

where n(t) is the neutron flux at any time, t, n(0) is the neutron density just before the step insertion was made, a_i is the fraction of post-shutdown neutron density which is attributable to the i^{th} delayed neutron group, and λ_i' is the apparent decay constant of the i^{th} delayed group. This parameter, λ_i', differs from the λ_i defined above only because of the sub-critical

multiplication of delayed neutrons due to new fissions which occur after shutdown. The more sub-critical the reactor is, the smaller the difference between λ_i and λ_i'.

An accurate interpretation of the dynamic performance of a highly enriched heavy water reactor, like the Georgia Tech Research Reactor, requires an accurate input of the parameters, delayed neutron group fractions, β_i; delayed neutron group decay constants, λ_i; and prompt neutron lifetime, ℓ^*, into these kinetics equations. Delayed neutron and delayed photoneutron parameters are available in the literature.[9,10,11] Because many of these data were obtained with small irradiated samples of uranium which do not necessarily represent an actual in-reactor situation adequately, they may not be entirely satisfactory for application to a specific reactor.

These small-sample experiments employ a piece of irradiated uranium to provide the delayed neutrons and high-energy gamma rays which can produce delayed photoneutrons. This sample is moved rapidly from the irradiation facility (a reactor or neutron generator) to a counting assembly, which incorporates heavy water, shielding material, and detectors. In a typical experiment, data are collected for several hours and stored for analysis. The experiment may be run with heavy water in place and then absent, in order to separate the photoneutron contribution. Alternatively, a bismuth shield may be placed around the sample to suppress the photoneutron-producing gamma rays.

Data obtained by the small-sample method have typically produced the accepted literature values for general kinetics applications. This technique was used in a number of "standard" delayed neutron experiments reported in 1947 - 1958.[9,12,13] By "standard" here we mean the delayed

neutrons which appear directly from the nucleus of a fission product atom as opposed to photoneutrons which are produced in an indirect manner. Accepted for many years were the parameters of Hughes, Dabbs, Cahn, and Hall.[9] More recently a comprehensive study of delayed neutrons has been carried out at Los Alamos by Keepin, Wimett, and Zeigler.[12] Their technique is typical of the small-sample method. A few grams of U-235 were irradiated in a neutron flux of the highest possible intensity for a period, the length of which depended upon whether short or long delayed neutron groups were to be emphasized. The use of a small sample minimized the multiplication of neutrons within the fissionable material itself. The combination of small sample and high flux produced satisfactory counting statistics without the necessity for large corrections for sub-critical multiplication. A pneumatic system transferred the irradiated uranium from the reactor to a counting tank equipped with detectors. The decay of delayed neutron activity was monitored by a crystal-controlled multichannel time delay analyzer with variable channel widths. The analysis of these data was accomplished by an iterative weighted least squares program on a digital computer. The best fit was obtained for six groups.

Small-sample methods have also been employed in the study of photoneutrons resulting from delayed fission product gamma rays above the 2.23 MeV threshold for interaction with heavy water. This reaction is described as: $\gamma + D \rightarrow n + p$. Spatz, Hughes, and Cahn[14] and Bernstein et al.[10] have used this method. Bernstein's work remains the apparent standard, although more recent work has been proceeding at Princeton[15] to obtain new measurements. Bernstein's apparatus was similar in concept to the Los Alamos equipment described above. The detection tank was a 19-inch diameter hollow

aluminum sphere which could be filled with D_2O. The sphere was immersed in a large tank of oil. The tank was a four-foot cube. A pneumatic rabbit containing the enriched uranium oxide sample could be transferred from the center of the reactor to the center of the sphere in about 0.25 second. The sphere could be filled with heavy water or drained, thus permitting separation of standard delayed neutron and photoneutron contributions. The neutrons were observed with fission chambers located at various radial positions in the oil bath. These detectors had a resolving time of 20 microseconds. The data were recorded by photographing the scaling circuit interpolation bulbs with an open shutter oscilloscope camera. A 0.1 second time signal was recorded on the film. As the film moved past the neon interpolation bulbs, each bulb produced a dashed line on the film for the time it was lit. Data were taken for approximately 70 hours. In this work some rather detailed corrections were required to account for the gamma-ray attenuation in the irradiation capsule of the pneumatic rabbit system and to extrapolate the data to an infinite D_2O volume. The accuracy of the results may suffer from this since the gamma energies are not now precisely known and were even less well determined at the time of this experiment in 1947.

In the Bernstein work, photoneutron group abundances were related to the yield of the 22-second delayed neutron group reported by Hughes,[9] and thus may be used to give group constants for kinetics calculations which are based on the best current estimate of that 22-second group yield. This is, in fact, what Finn did in comparing experimental and predicted reactivity values for the Savannah River Process Development Pile reactor.[16] An additional long-lived precursor, obtained from absolute radiochemical measure-

ments has been reported by Ergen[17] and this group is frequently included in recommended group constants for kinetics calculations.[11]

Because of the experimental method employed and the corrections applied to obtain these group constants, they do not take into account the gamma-ray attenuation tending to diminish the number of gamma rays which are above the energy threshold when they attain access to the heavy water of a given D_2O reactor. Neither do they incorporate the potentially higher effectiveness of delayed neutrons in producing subsequent neutron generations due to their lower mean energy spectrum.[18] In order to assess the adjusted values which apply to a specific reactor, delayed neutron parameters have been determined by analyzing the post-shutdown flux in that reactor.

At the Canadian low-power ZEEP reactor, Johns and Sargent[19] analyzed the decaying flux over a 37 hour period. Their abundance values were related to that value obtained for a prominent group whose period corresponded to one found by Hughes et al.[9] In their work, carried out in 1946-1947 and published in 1953, three proportional counters of different sensitivities were used. Each was placed at a different location in the reactor. To obtain the flux as a function of time over the periods of operation and shutdown, an inter-calibration of relative sensitivity of the counters was accomplished. All relative neutron fluxes were then expressed in terms of that of the most sensitive neutron counter. Counts, in units of 128, were recorded on moving paper by pen recorders. No assessment of system resolving time per se is given, but the statement is made that surprisingly large counting losses were probably caused by an unnecessarily long dead time in the discriminators. However, it is also

noted that, for the counting data on which most of the decay analysis depends, the losses were five percent or less. The ZEEP reactor was operated at power levels up to 50 watts for periods of 10, 50, and 230 minutes before shutdown and the resulting parameters were corrected from these periods of operation to represent infinite operation.

Tunnicliffe[20] performed a similar experiment in 1952, collecting data with ionization chambers in the NRX reactor over a 24 hour period. In his analysis, however, Tunnicliffe assumed the validity of the decay constants of the six standard delayed neutron groups as determined by Hughes et al.[9] and fitted just the photoneutron values by a least squares method. In the weighting of data points, the standard deviation, σ, was obtained by attributing a reading accuracy to the chart records of one percent of the full scale deflection and neglecting any systematic errors from range to range of the D.C. amplifiers.

At the Swiss Federal Institute for Reactor Studies, Würenlingen, data on DIORIT have been taken,[21] but as yet no results have been published. DIORIT, ZEEP, and NRX are all natural-uranium fueled and, therefore, are not typical of highly enriched research reactors either in dimension or gamma-ray attenuation. There is the additional complicating factor of an appreciable fast fission contribution in these reactors with the resulting delayed neutrons from a different fissionable species, U-238. The ratio of fast to total fissions in DIORIT amounts to about five percent.[21]

All of the work which has been described in the preceding pages has dealt with U-235. Summaries of detailed neutron experiments for many fissionable species have been prepared by Keepin.[22,23]

The technique of analyzing the post-shutdown flux in an actual reactor has not been applied to a highly enriched heavy-water reactor, such as the Georgia Tech Research Reactor. Consequently, the appropriate parameters required in the various mathematical treatments of reactor kinetics for this type reactor have not been determined. The flux decay analysis procedure has the advantage of deriving parameters from experimental operation of precisely the same reactor type for which the results will be utilized, thus assuring their applicability. In practice, one fits the data collected following a rod drop to a sum of terms, $\sum_{i=1}^{M} a_i e^{-\lambda_i' t}$, representing a linear superposition of the exponential decay periods of the delayed neutron activity present at the time of shutdown. The results may be examined in order to determine how many delayed neutron groups, M, are required to represent this flux decay process. The intermediate parameters obtained from the data fitting analysis must be adjusted to correct for the sub-critical neutron multiplication (or more specifically, the production of new fission products which is initiated by delayed neutron induced fission) and for non-equilibrium of precursors. This is required in order to remove from the parameters any dependence on power history and degree of sub-criticality. These corrections are obviously necessary to ensure the general utility of the results; they are also prerequisites to comparing them with literature values which have been obtained using small irradiated uranium samples with a negligible sub-critical multiplication and no appreciable irradiation history. Such a comparison is useful in itself, since it sheds light on the mechanisms operating within the nuclear reactor which cause the differences between the small-sample values, β_i, and the effective values, $\beta_i$eff.

The differences in these parameters may be attributed to several causes. First, since the energy spectra of delayed neutrons and delayed photoneutrons are known to be lower in average value than those of prompt neutrons, the delayed neutrons require less time to slow to thermal energy, hence experience less capture and leakage in the process. According to the data of Batchelor and Hyder,[18] the average energy of the six standard delayed neutron groups is 430 keV. The photoneutron spectrum is dependent on the spectrum of gamma rays producing the reaction; in a reactor, most known precursor gamma rays possess energies less than three MeV and hence would produce neutron energies less than about 0.8 MeV in a D_2O reactor. This may be compared with the 2 MeV average energy of U-235 fission neutrons.[24] This gives delayed neutrons a greater importance as far as their ability to initiate subsequent generations of neutrons, and this greater "worth" results in an effective abundance fraction which, for some reactors, has been 25 percent[25] greater than the fraction which one might expect from data obtained with a small sample. It is evident that the magnitude of this energy-dependent worth is a reactor-dependent quantity since slowing down and leakage vary among reactor types.

Competing with the enhancement effect just mentioned is another effect which is a strong function of the configuration of fuel elements used. Since photoneutrons in heavy water are the result of an interaction of a gamma ray possessing more than the 2.23 MeV threshold energy with the deuterium of the heavy water, and these energetic gamma rays originate in the fuel, any interaction which a gamma ray undergoes between its formation and its arrival in the heavy water region tends to result in its removal or reduction in energy below threshold value. These events cause a

smaller effective abundance fraction for delayed photoneutrons than one might observe in the small-sample method where very little attenuation occurs in the uranium sample and cladding. In the Process Development Pile at the Savannah River Plant, gamma-ray interactions with materials other than D_2O result in a reduction of the intensity of the high-energy gamma rays by 78 - 82 percent of the initial level.[16] In the Zero Energy Experimental Pile at Chalk River, the comparable value was calculated to be 74 percent.[19] At DIORIT the value 64 percent is used.[21] In the Massachusetts Institute of Technology Reactor, the removal or reduction was calculated to be 31 percent.[26]

As a result of these factors, the effective abundance fractions for a given reactor may be significantly different from those which apply to another reactor or those which might be obtained from small samples. A good illustration of the effect which choice of kinetics parameters may have on such practical considerations as reactor control rod calibrations or other reactivity assessments appears in Finn's work.[16] His plot of reactivity versus reciprocal doubling time is reproduced in Figure 1. It demonstrates the large discrepancy which exists for a particular reactor, PDP, between the experimentally observed relationship and those calculated using the indicated parameters and parameters adjusted for gamma transmission. Variations of this magnitude indicate the desirability of assessing the delayed neutron and photoneutron kinetics parameters for each specific reactor type.

Purpose of the Research

It was the purpose of the research described in this thesis to:

1. Establish the number of delayed-neutron and photoneutron groups which are required to describe accurately the kinetics behavior of the

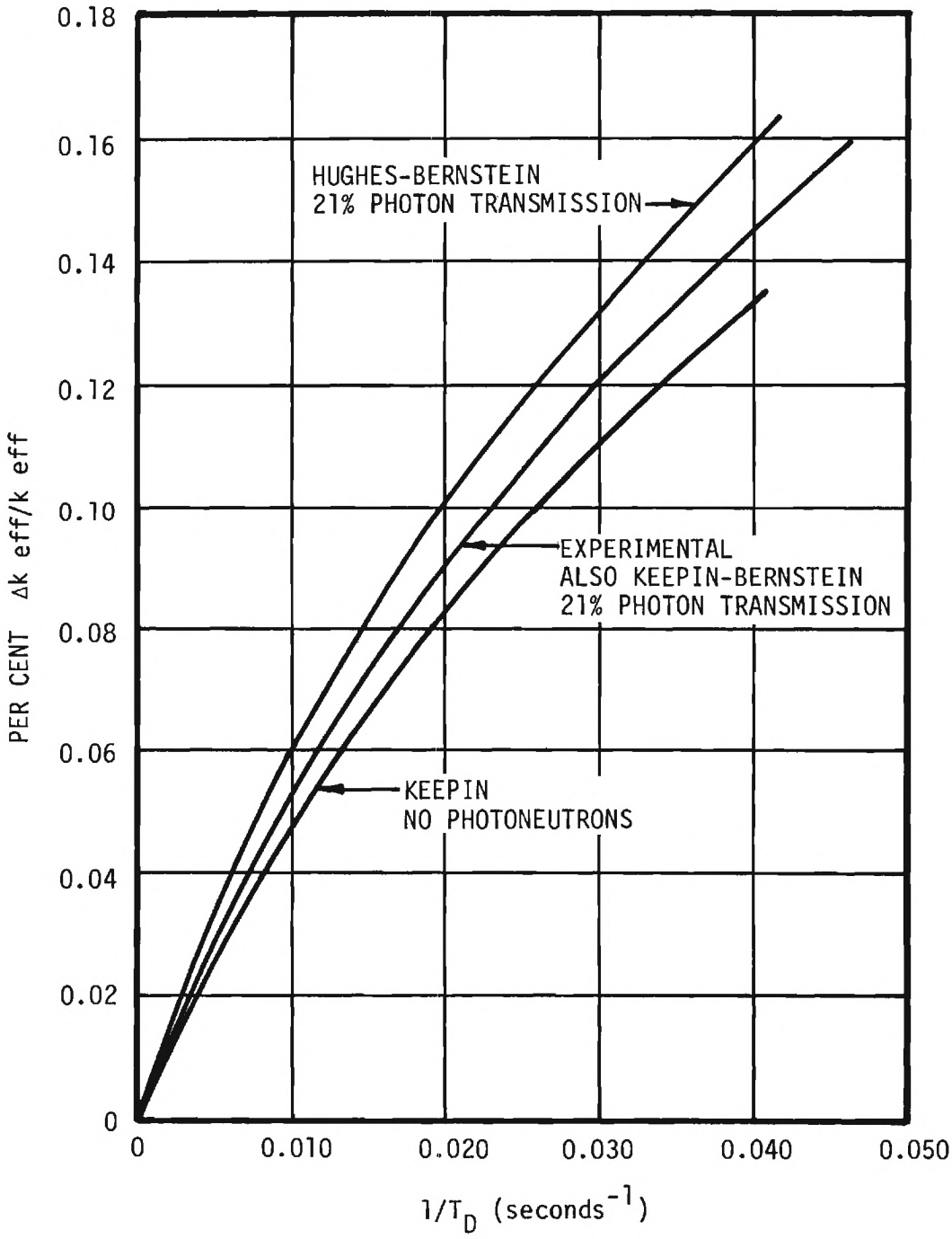


Figure 1. Process Development Pile -- Reactivity and Reciprocal Doubling Time Relationships for Various Kinetics Parameter Sets (after Finn[16])

Georgia Tech Research Reactor.

2. Determine the relative abundance, $\beta_i eff/\beta eff$, of each resolved delayed neutron and photoneutron group.

3. Determine the decay constant, λ_i, of each resolved delayed neutron and photoneutron group.

4. Compare the results obtained above with current literature values.

The delayed neutron and photoneutron kinetics parameters which are the product of this research will permit more precise kinetics calculations for highly enriched heavy-water reactors in general. In addition, some of the techniques that have been developed to facilitate the accomplishment of the purposes stated above may find uses in reactor calculations which are not restricted to a specific reactor type.

CHAPTER II

INSTRUMENTATION AND EQUIPMENT

The Georgia Tech Research Reactor

The Georgia Tech Research Reactor is a heterogeneous, heavy-water-moderated and cooled research reactor located at the Frank H. Neely Nuclear Research Center on the campus of the Georgia Institute of Technology. This reactor is fueled with highly enriched (greater than 93 percent U-235) plates of an aluminum-uranium alloy that is aluminum clad. The reactor is designed for one megawatt operation with an associated thermal neutron flux in excess of $10^{13} n/cm^2$ sec. At the time of the research reported here, low power testing and calibration were in progress and the power level was limited to 200 watts in order to facilitate handling of irradiated fuel elements during flux mapping and other reactor physics measurements.

While the fully loaded reactor core might contain as many as 19 fuel elements spaced six inches apart in a triangular array about two feet in diameter and two feet in height, the data for the present research were obtained on a 13-element core. Figure 2, a horizontal section of the Georgia Tech Research Reactor (hereafter to be referred to as GTRR), illustrates the positions of experimental facilities and fuel elements.[27] The 13-element core occupied locations V2, V4 - V7, V9 - V11, V13 - V16, and V18. The fuel element is illustrated in Figure 3.[27] Each standard element contains ten curved fuel plates 0.060 inch thick and 23.5 inches long. The aluminum cladding is 0.020 inch thick on the sides and 0.205 inch over the

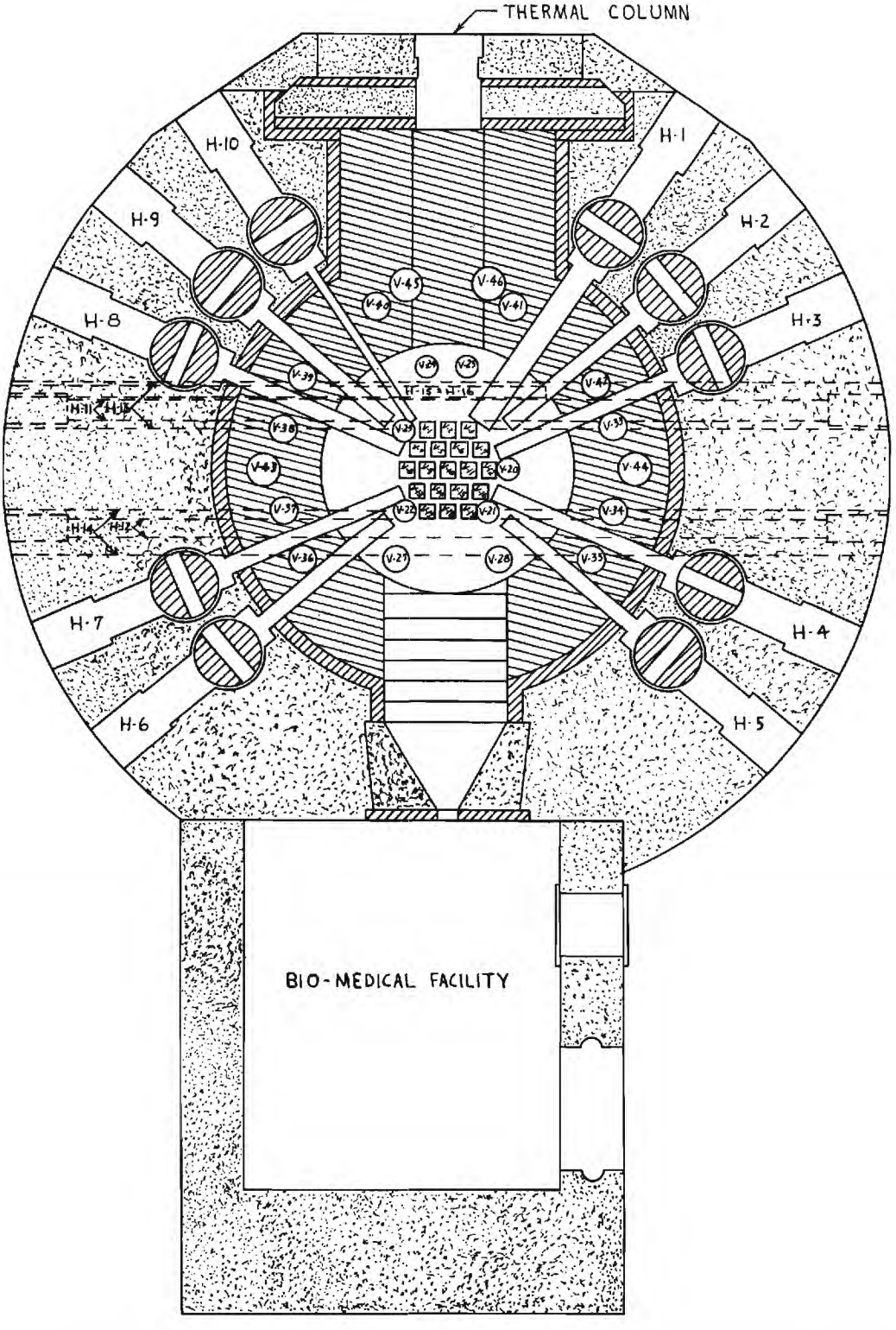


Figure 2. Georgia Tech Research Reactor -- Horizontal Section at the Core Mid-plane

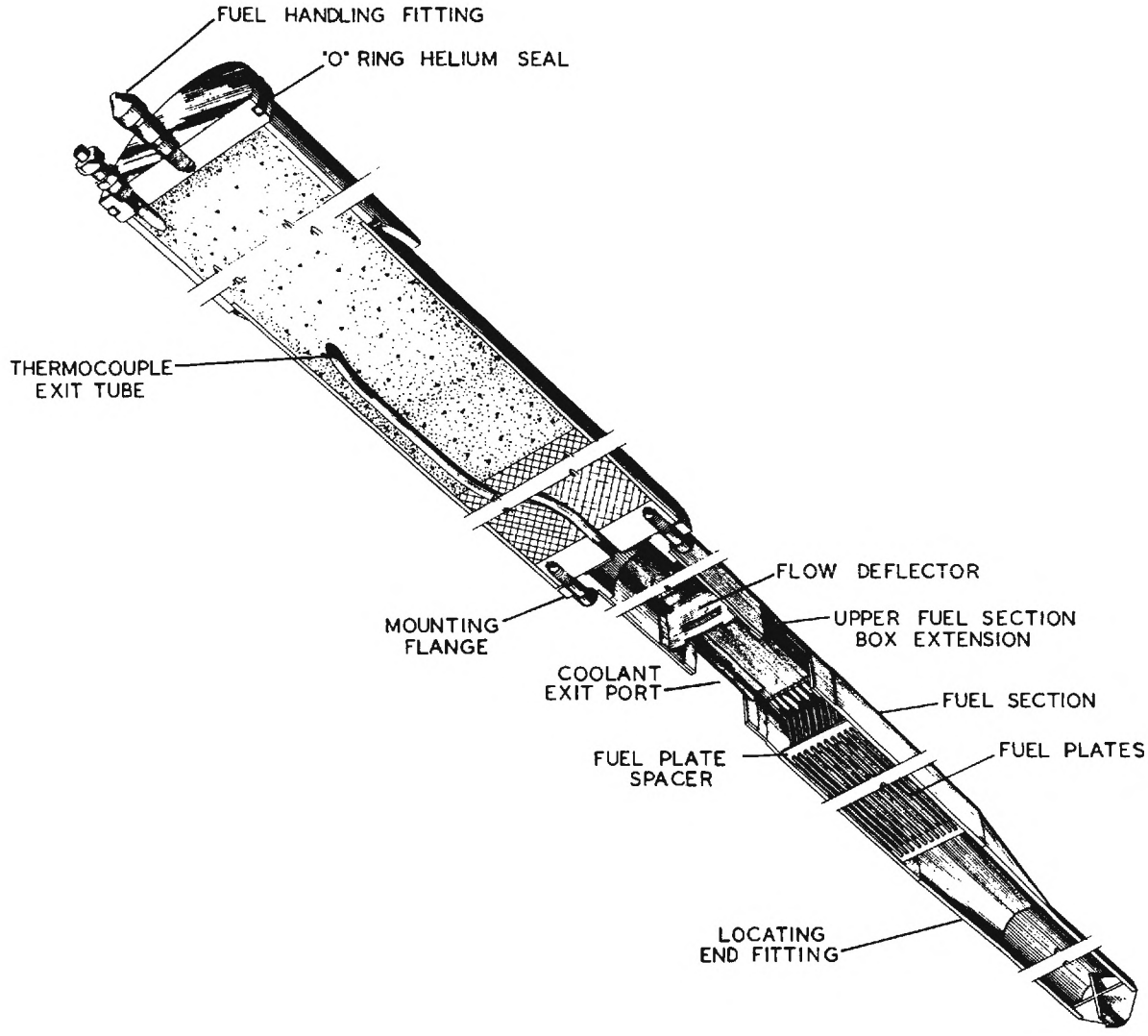


Figure 3. Georgia Tech Research Reactor -- Fuel Element Cutaway View

edges. Coolant passes in the 0.183 inch space between adjacent fuel plates. Each fuel element contains 142 grams of U-235.

The core is contained within a 6-foot diameter aluminum vessel and is immersed in heavy water so as to provide a two-foot thick layer of D_2O reflector. Figure 4, a vertical section,[27] illustrates the placement of the reactor vessel within the graphite reflector and the concrete biological shield. This figure is also marked to indicate the location of horizontal through-tubes designated H11 and H12. These experimental facilities, which pass tangent to the core, were used for the placement of the neutron detectors of this experiment.

Figure 5 is a photograph of the reactor vessel interior prior to fuel element loading. This picture gives a good view of the semaphore blade type shim safety rods. These blades consist of cadmium metal 0.040 inch thick, clad inside and outside with type 6061 aluminum alloy. They weigh about 20 lb. each and are 5.5 inches wide by 1 inch thick. The cadmium section is 45.5 inches long with the total length being 60 inches. The full travel of these safety blades is 55°. Also visible in Figure 5 are the aluminum thimbles in which the through-tubes H11 and H12 pass.

Detection and Data Recording Systems

In applying the method of reactor flux decay analysis for obtaining kinetics parameters, it is essential that reliable neutron detectors be employed in data collection. Accuracy of the parameters which result from the analysis is enhanced if the measurements of neutron flux level in the reactor over various time intervals are made at the highest possible count rates, since this contributes to the statistical precision of the data.

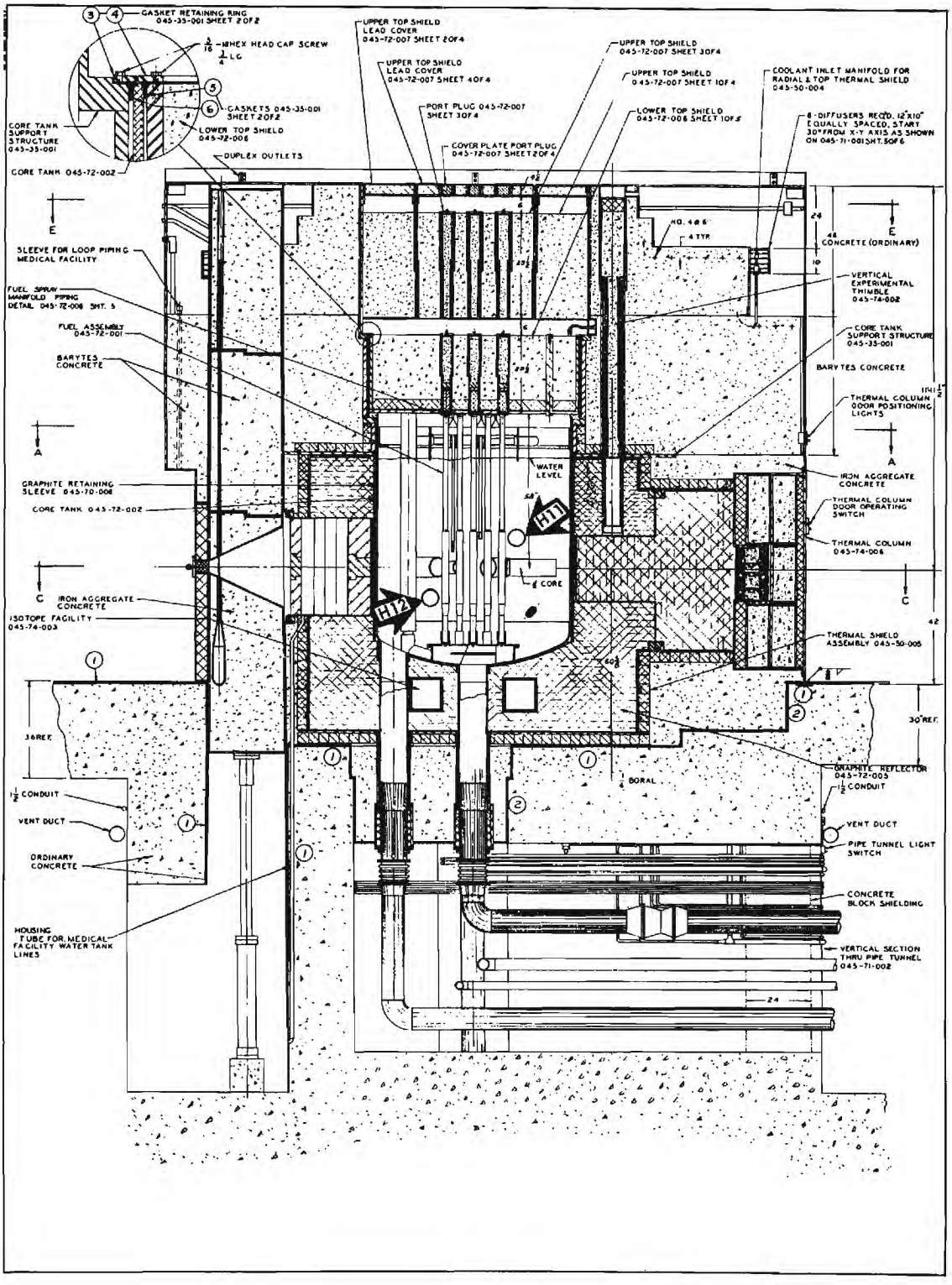


Figure 4. Georgia Tech Research Reactor -- Vertical Section

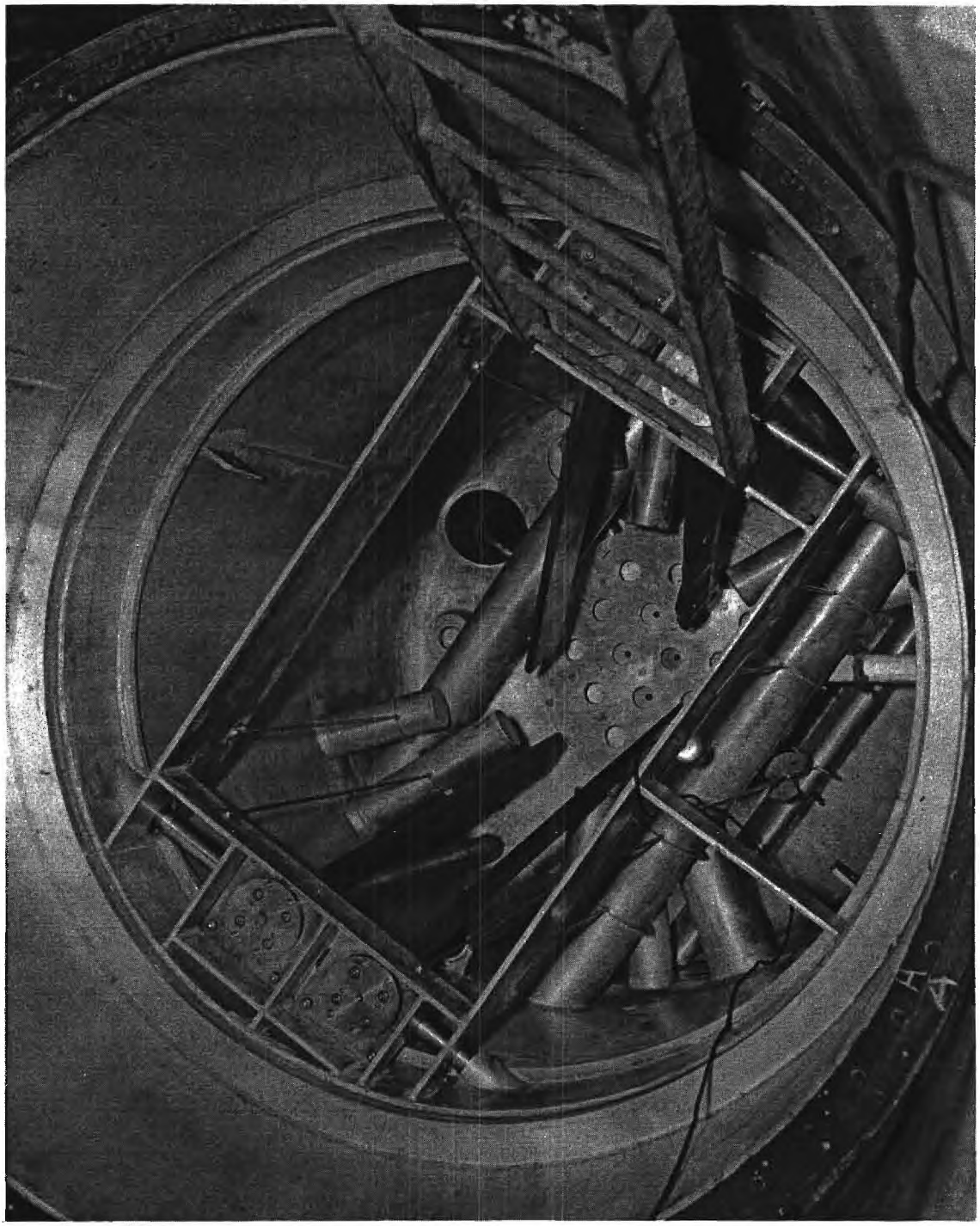


Figure 5. Georgia Tech Research Reactor Vessel Interior Before Fuel Element Loading

In the present work, the flux was monitored over a several-day period. Therefore, we required detection systems which were stable at both the initially high count rates immediately after the reactor shutdown by rod drop and at the final low count rates. As the count rate diminished in the course of the experiment, the data were obtained and stored with decreasing frequency. The interval over which pulses from the detectors were accumulated before storing for subsequent recording varied from 0.1 second to 100 seconds.

Two different types of detection systems were used in this work in order to provide a check on the consistency of results. The concept of the first, which uses a fission chamber detector, was suggested by Behringer.[21] This system was supplied by Reactor Controls, Inc. in accordance with our specifications. The second system was developed by the Instruments Group of the Nuclear Research Center. It incorporates a scintillation element. This particular approach of providing two different detector types was chosen since both types appeared to offer attractive possibilities for short resolving times well under a microsecond, and yet both were developmental in nature. Figure 6 is a block diagram of the major components of these two detection systems.

Fission Chamber System

The fission chamber, designated as Reactor Controls, Inc. Series RC-26-2451A-1T, has been specially designed for very high counting rates and for high neutron fluxes and gamma-ray fluxes. It is illustrated in Figures 7 and 8. The detector is one inch in diameter with a neutron sensitive length of four inches. The fission chamber has a neutron sensitivity of 0.05 to 0.07 counts/nv. Its construction is all aluminum except

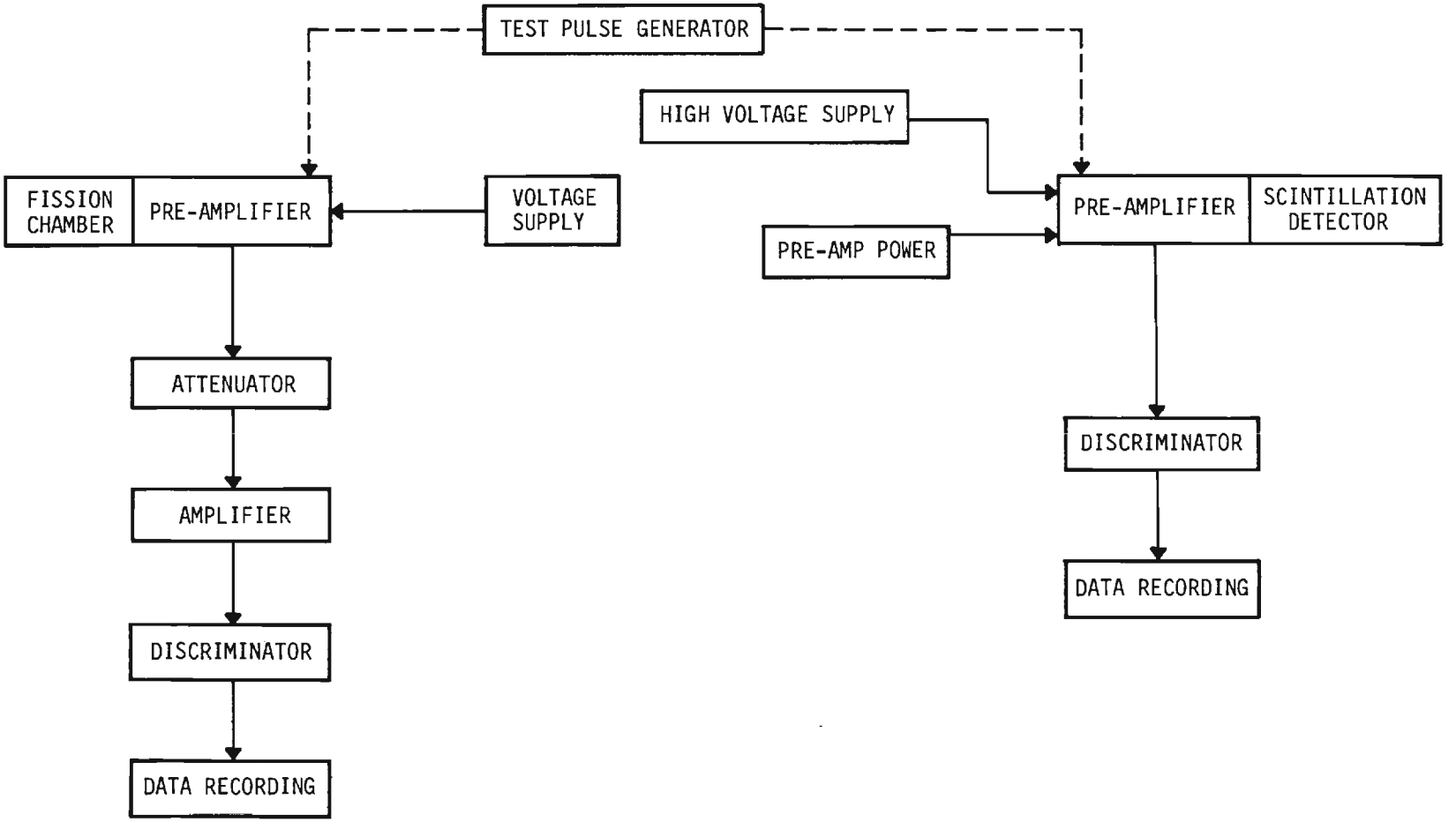


Figure 6. Block Diagram of Neutron Detection Systems

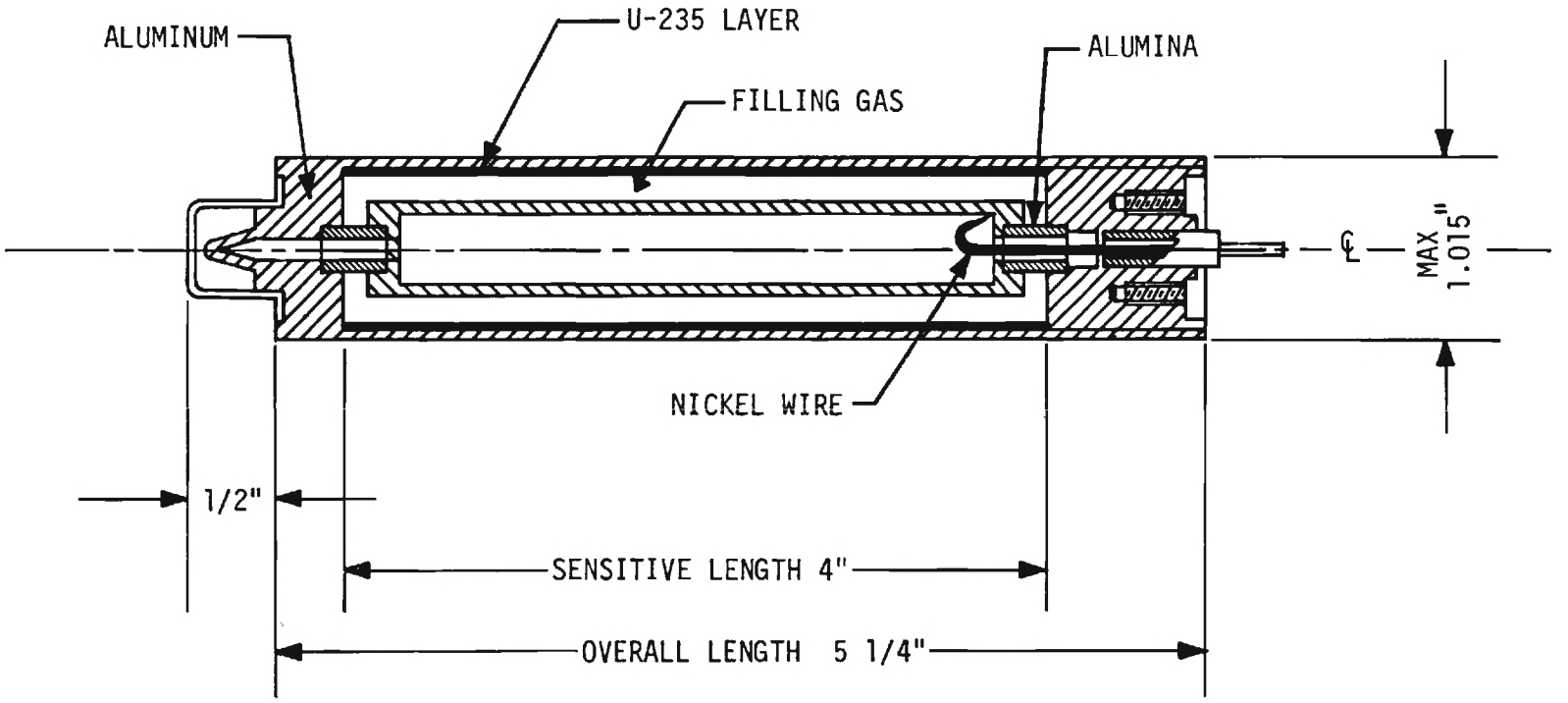


Figure 7. Cross Section Diagram of Fission Chamber

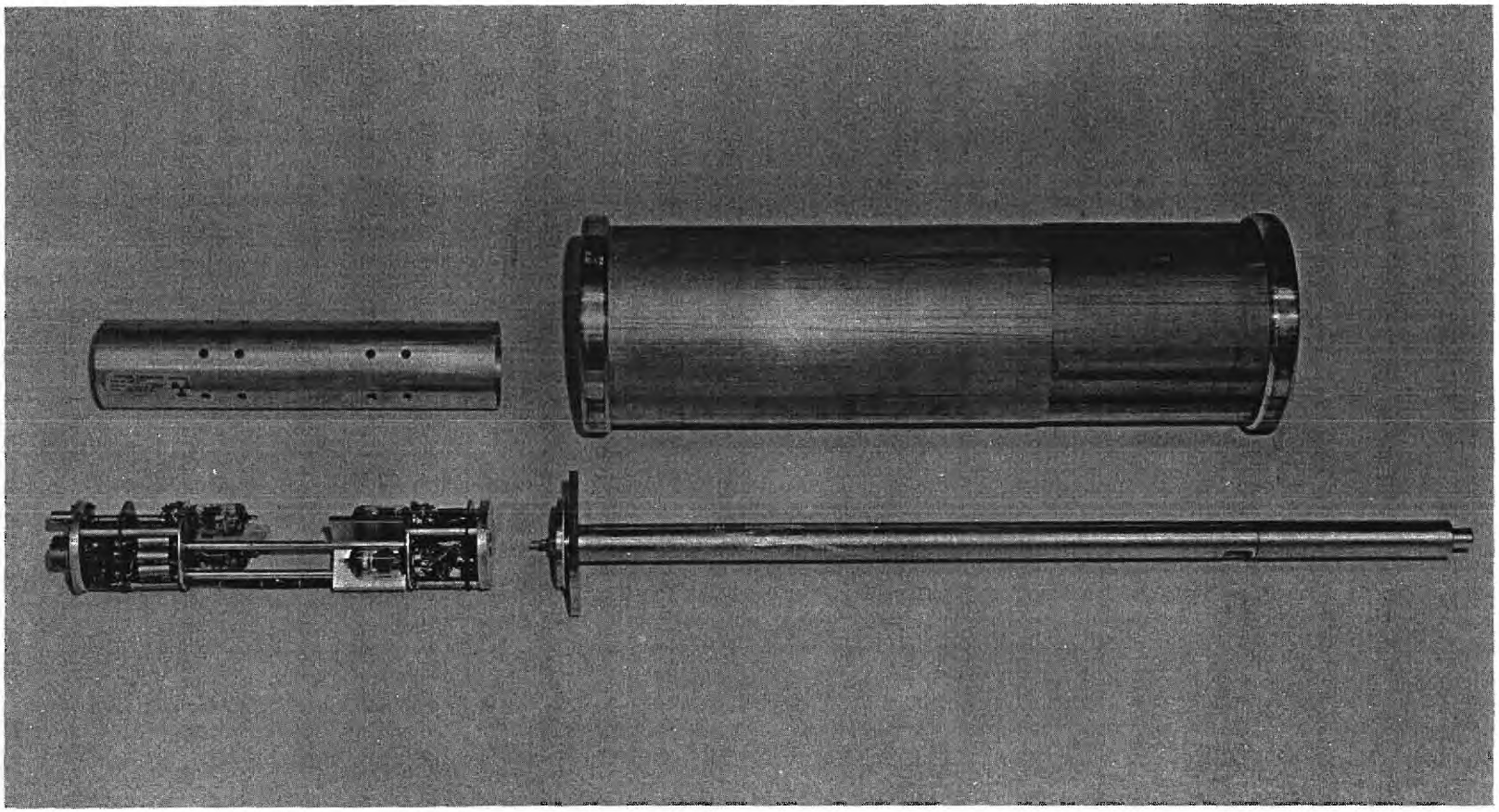


Figure 8. Fission Chamber, Radiation Shield, and Preamplifier

for an alumina ceramic-to-metal seal.

In order to reduce the neutron and gamma radiation at the preamplifier while maintaining the detector in a high neutron flux, such as may be obtained in a radial beam port or through-tube of the GTRR, a 20-inch long solid aluminum plug provides shielding between the detector and the preamplifier. The plug fills the inside diameter of the beam hole to minimize streaming. A 1-inch diameter hole through the large-diameter shield plug accepts a 1-inch diameter shield plug. The detector is mounted on this 1-inch diameter, 19-inch long aluminum shielding plug, and this in turn mounts on the preamplifier. Signal connection from the detector to the preamplifier is made by a 1/16 inch diameter nickel rod mounted on spaced ceramic beads inside a 1/4 inch diameter hole through the 1-inch diameter shielding plug.

The preamplifier was designed by Reactor Controls, Inc. for random nuclear pulse amplification with double differentiation pulse shaping. Double pulse resolution is 240 nanoseconds. The preamplifier has an overall gain, exclusive of pulse shaping stages, of greater than 600, with a rise time of less than 20 nanoseconds. Gain is stabilized by precision resistors in the feedback loops around all sections of the preamplifier.

The input stage is a very low noise cascode circuit designed for random pulse inputs from 50 microvolts to 2 millivolts, negative peak. Provision is made for supplying + 500 volts to the fission chamber signal electrode. First differentiation is accomplished by RC circuitry between amplifier sections.

The cathode follower output of the preamplifier drives a 95 ohm balanced line cable at signal levels up to 1 volt peak. This high level

output is desirable for use with long cable lengths without pickup of significant noise. Second differentiation is accomplished at the main amplifier location after the long cable length. A schematic diagram of the preamplifier may be found in Appendix G.

The preamplifier is packaged in a 2-1/2 inch diameter cylinder, 12 inches long. Fission chamber input is at one end, power and signal cables are located at the opposite end. All components have a low temperature coefficient and, except for the connectors and cables, are capable of operation at temperatures up to 250° F.

An attenuator panel receives a 95 ohm balanced line input and converts it to a 50 ohm single-ended line. The signal is passed through a toggle switch attenuator, adjustable over 42.5 db by 0.5 db steps, and is then RC-differentiated at the output at 50 ohms impedance. The differentiated signal is amplified by a gain-of-ten amplifier having a rise time capability of 2 nanoseconds.

A pulse amplitude discriminator produces a 0.5 V amplitude standard pulse at 50 ohms impedance for each input pulse with amplitude greater than 0.10 V. The pulse amplitude discriminator is capable of less than 10 nanoseconds double-pulse resolution. Both the amplifier and discriminator described above are modules manufactured by Chronetics, Inc.

The preamplifier is powered by a Kepco Model KR-2 regulated power supply set at 212 V. The supply is modified to provide decoupling between preamplifier stages by three RC filters and to provide the fission chamber polarizing voltage regulated by six 85A2 regulator tubes in series. A separate 6.3 V supply provides D.C. filament power to the preamplifier tubes. A connector is provided for the seven-wire cable going to the pre-

amplifier. The power supply is typically located within 15 feet of the preamplifier to reduce filament voltage loss and noise pickup.

The fission chamber system as a whole may, of course, be made sensitive to the alpha rays which are emitted by the fissile uranium coating of the chamber, by sufficient amplification. This provides a convenient test source. But the system possesses an effective alpha discrimination capability without appreciable loss of neutron sensitivity. This discrimination capability may be observed in Figure 9. In Curve A, the detector was tested in the reactor up to a relative gain setting of 61. When the detector was removed from the reactor, there were no counts at this gain setting, and, as indicated in Curve B, it was necessary to increase the relative gain to about 67 before it was possible to observe the lower energy alpha source present within the active material of the fission chamber itself. Thus, it may be concluded from Figure 9 that the gain necessary to obtain a nearly flat neutron flux response is well below that required to detect even a minimum background contribution from alpha-produced pulses.

Scintillation Detection System

The active scintillating element used in these detectors was a Nuclear Enterprises, Ltd., Model NE-421 slow-neutron sensitive button of one inch diameter, shown in Figure 10. It employs a lithium compound whose Li content is enriched to 96 percent Li-6 dispersed in a ZnS matrix 0.025 inch thick. The detector design is a modification by Nuclear Enterprises of the type reported by Stedman.[28] The manufacturer specifies that this element provides 50 percent detection efficiency for thermal neutrons (0.01 eV or less) and gamma discrimination to the extent that thermal neutrons may be effectively measured in the presence of gamma radiation as high as

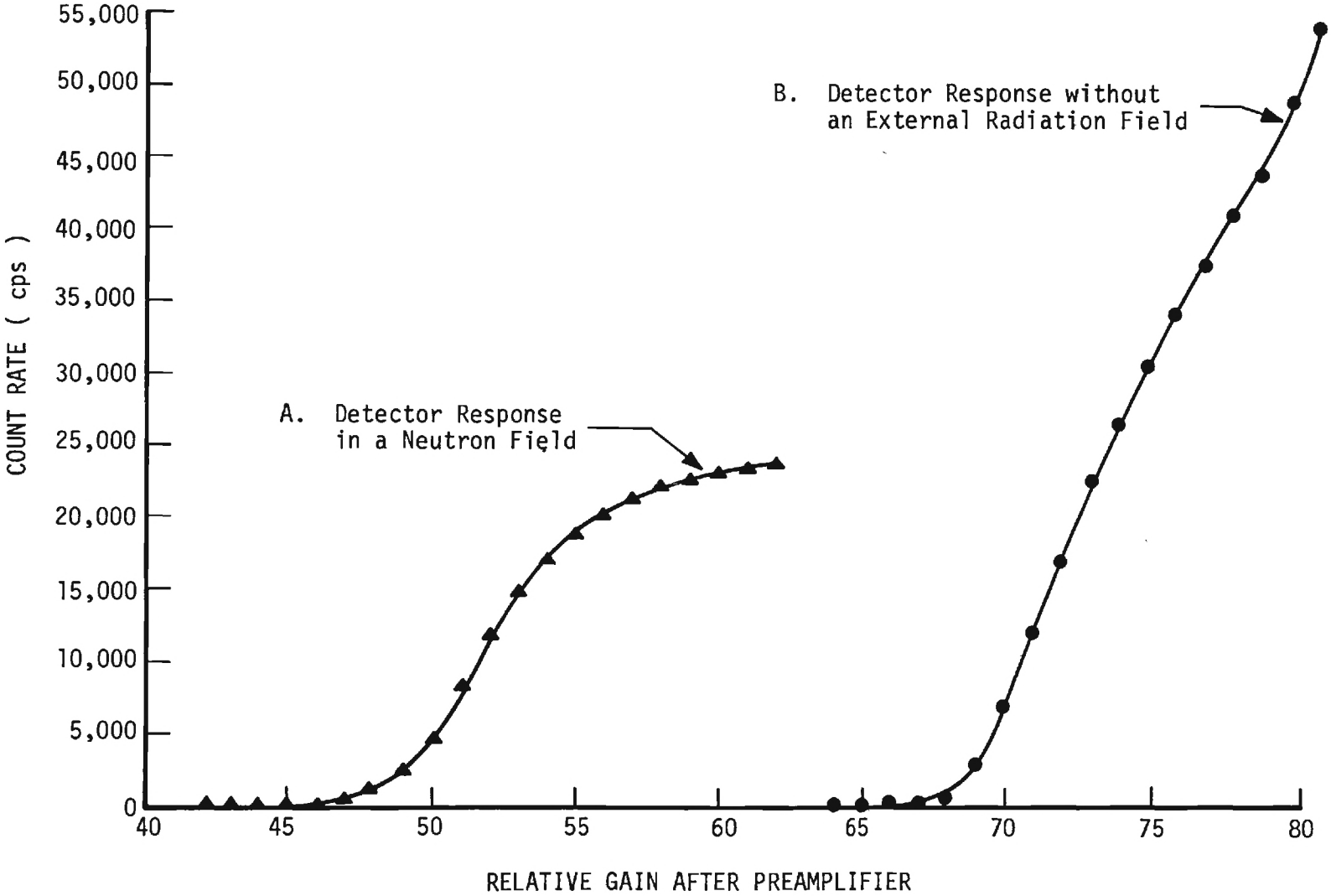


Figure 9. Performance Characteristics of Fission Chamber Both in and out of the Reactor as a Function of Preamplifier Gain

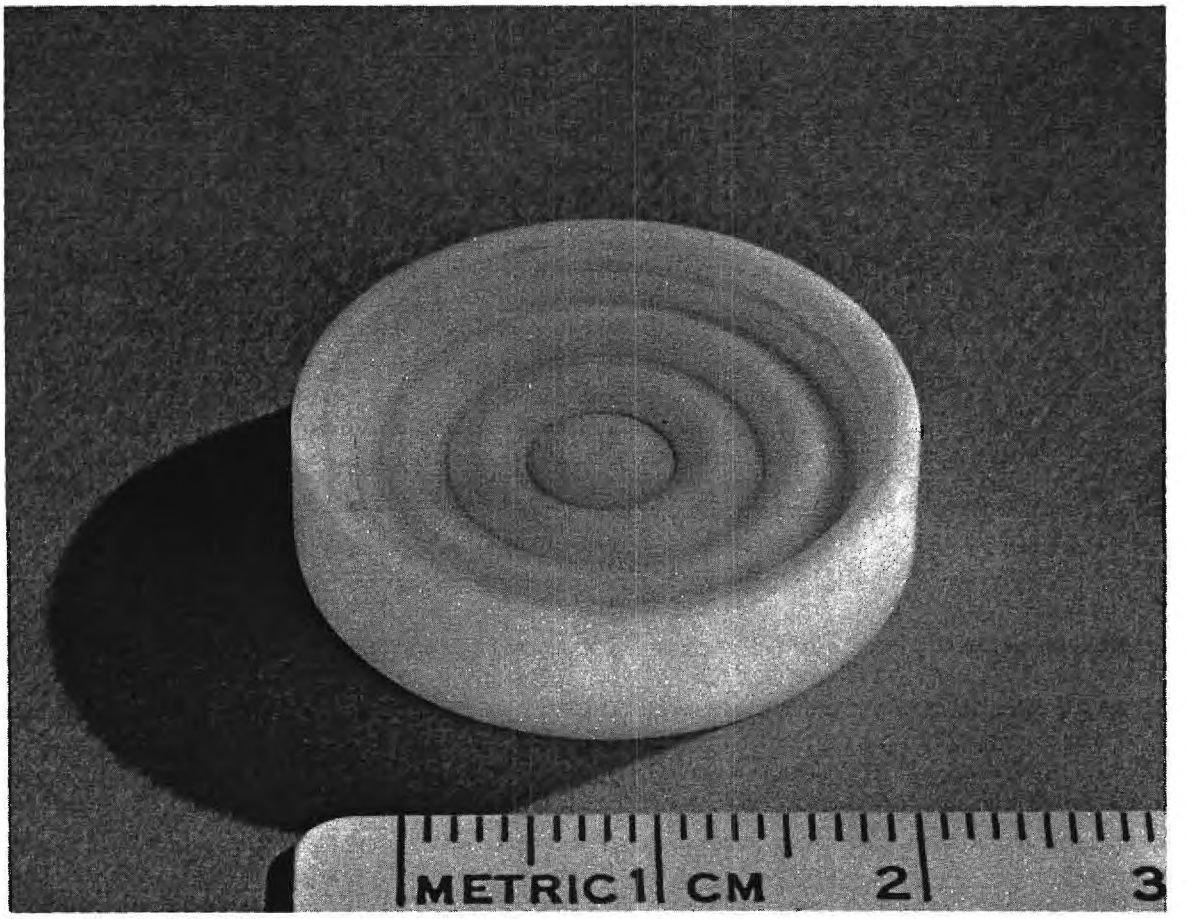


Figure 10. Scintillation Detector Active Element

10^7 gamma rays per neutron.

The high voltage was supplied to the photomultiplier tube by an NJE Corporation Regulated High Voltage D.C. Power Supply, Model No. S-327. This unit has an output range of 500 - 5000 volts, 0 - 10 mA. According to the manufacturer's specifications, it has a maximum ripple of 5 V, a static regulation of ± 100 mV (obtained with constant input voltage and load changes from a nominal half load to no load or full load, respectively), or ± 0.01 percent (obtained with constant load current and input changes from a specified nominal A.C. Voltage to + 10 percent voltage or - 10 percent voltage, respectively), and stability of 0.03 percent per hour or 0.06 percent for an eight hour period.

The photomultiplier tube employed in the scintillation detector was an Amperex XP1010 tube with ten stages and a cesium-antimony semi-transparent flat cathode of 32 mm diameter. This tube was selected for its low background; the manufacturer specifies a room temperature dark current of 10^{-15} A/cm^2. The photomultiplier tube was obtained from the Harshaw Chemical Company in an integral assembly incorporating the Nuclear Enterprises NE-421 scintillation detector and mu-metal magnetic field screening cylinder in a light-tight jacket.

The voltage divider string is built up in a standing resistor design as illustrated in Figure 11. This type of construction, along with the detector casing ventilation, permits heat removal by natural convection. At the operating level of 1150 volts, the photomultiplier draws 2.8 milliamperes of current. The high current drain of this voltage divider string is an intentional result of the design of the string with relatively low values of resistance. This was done in order to minimize any effect which

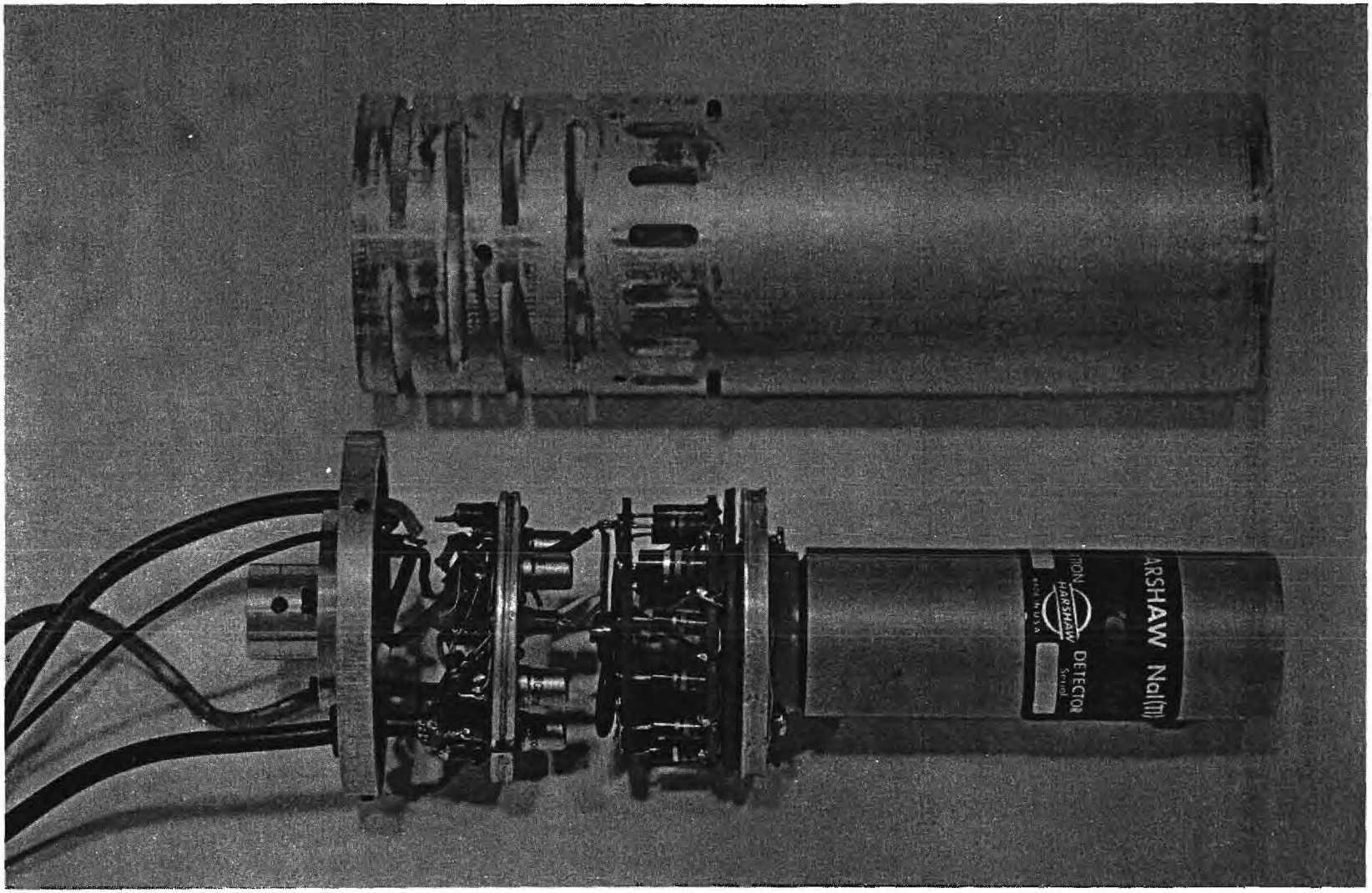


Figure 11. Detector Incorporating Neutron Scintillator, Photomultiplier Tube, and Preamplifier

a rapid change in count rate might have on the gain of the assembly.

The preamplifier used in the scintillation detector was designed and built by the Nuclear Research Center Instruments Group, directed by R. E. Meek. This preamplifier provides double RC differentiation and two stages of amplification. The input stage from the photomultiplier tube is a cathode-coupled amplifier from the photomultiplier tube with cathode-follower input and grounded-grid-amplifier output to the second stage. The first stage has a gain of 15. The second stage is a White cathode-follower with a fractional gain of 0.9. A schematic diagram of this preamplifier circuit is included in Appendix G.

Heat dissipation within the preamplifier assembly was enhanced by mounting the Nuvistor tubes in a 1/4 inch aluminum chassis which was in turn attached directly to the sides of the aluminum detector enclosure. The output pulse width of this preamplifier has been observed by oscilloscope to be 0.4 microsecond with actual input pulses from the phototube. The pulse risetime is 40 nanoseconds. The preamplifier has a test pulse input connection.

Data Recording System

Technical Measurement Corporation 400-channel pulse height analyzers were operated in the multiscaler mode in order to provide continuous accumulation of data on two parallel detection channels. Three multiscalers were required; one Model 401 and two Model 404 units were available.* These units were modified by the Nuclear Research Center Instruments Group in order that they might operate as two megacycle scalers. The electronic modifications

*One TMC analyzer was supplied on loan through the courtesy of the TMC-Ellison Division, Atlanta, in connection with the neutron generator project.

required to provide dwell times beyond the inherent capability of the TMC instruments, and to furnish experiment start pulses and signal routing were all assembled external to the multiscaler units themselves so as to preserve their general utility as pulse height analyzers. Appendix G includes a block diagram of the portion of this modification which accomplished multiscaler sequencing, if required, as well as other schematic diagrams of the electronic circuitry used to amplify the capabilities of the TMC analyzers.

The data recording equipment is capable of automatically sequencing from one analyzer to the next, although this feature was not used in the tests reported in this thesis. Sequencing of multiscalers is based on a signal appearing at the readout light of the unit which has filled its required number of channels. In this mode of operation, data readout is accomplished automatically in order to prepare the analyzer to receive more data. Since the analyzer has been programmed for automatic readout of data by positioning the front panel control switch on "automatic," a readout indicator light appears at the instant the last of the 400 channels is gated off by the "dwell advance" clock. The "automatic" programmed operation would normally result in the analyzer returning to the accumulation of data after completing readout. Since this is not desired, it is necessary to inhibit this step. This is accomplished by routing the same pulse which turns on the readout light to the "accumulate control" section and using it to appropriately modify the state of the control elements. Simultaneously, the readout pulse appears at the "accumulate control" section of the succeeding multiscaler and permits it to initiate the "accumulate" action. This consists of giving a command to the multiscaler to accept counts in its first channel when it receives the first dwell advance pulse. It receives that

dwell advance pulse about one millisecond later, since the "start-accumulate" command also goes to the clock via a built-in delay that is provided to ensure full channel width in time. The timing of subsequent channel advance pulses depends upon the dwell time per channel which has been selected on the control panel. This timing may be provided by either the same or separate circuits for each multiscaler and is derived from a base clock having a 100 kc crystal oscillator as its prime mover. This oscillator is divided down by two stages of five each and one stage of four so that the base clock output frequency is accurately 1.0 kc. The clocks associated with each multiscaler are themselves unijunction transistor oscillators which further divide the one kc base clock input by factors of ten in order to obtain dwell advance pulses every 0.01, 0.1, 1, 10, 100, and 1,000 seconds. With adjustment of variable resistor component values, still other dwell times may be obtained. The accuracy of the resulting dwell advance pulses is typically better than one part in 10,000, as measured by a Hewlett-Packard Model 5233L electronic counter. Channel advance time, i.e. time from the end of storage in one channel to the beginning of storage in the next, is 34 μsec. While unijunction transistor oscillators have adequately provided the timing required in this experiment, they are not recommended for this application which demands consistent performance during a several-day experiment. Our experience has indicated that there are component-matching and tuning requirements which make the initial and post-repair calibrations extremely tedious.

Another method of starting multiscaler accumulation is by use of an external signal, or start pulse. In these experiments, all three analyzers were started either manually before rod drop or automatically at the time

shim safety rod #3 engaged its lower limit microswitch. A 110 V signal is available at the control rod drive position when this occurs, and it is brought to the control panel and shaped so as to be suitable for actuating the "accumulate control" section of the appropriate multiscalers.

The actual signal inputs from the detection system pulse height discriminators are brought to the control panel via BNC connectors and wired to multipole switches for routing to the appropriate multiscaler inputs. The control panel and other systems may be seen in Figure 12.

Data readout from the multiscaler memory was accomplished by punching paper tape. Two Tally Model 420 Tape Perforators were used. These units punch at a rate of 60 characters per second and produce six channel tape in a 1-2-4-8-X-0 code. Parallel-to-serial conversion of data from the multiscaler memories is accomplished with TMC Model 520 Punch Control units. These provide a capability for manual input of coded punches on the tape. This feature was used preceding each readout in order to identify the test number and sequence for the associated data. With the multiscaler "reset" toggle switch set on "automatic," the memory may be automatically cleared during the readout to paper tape.

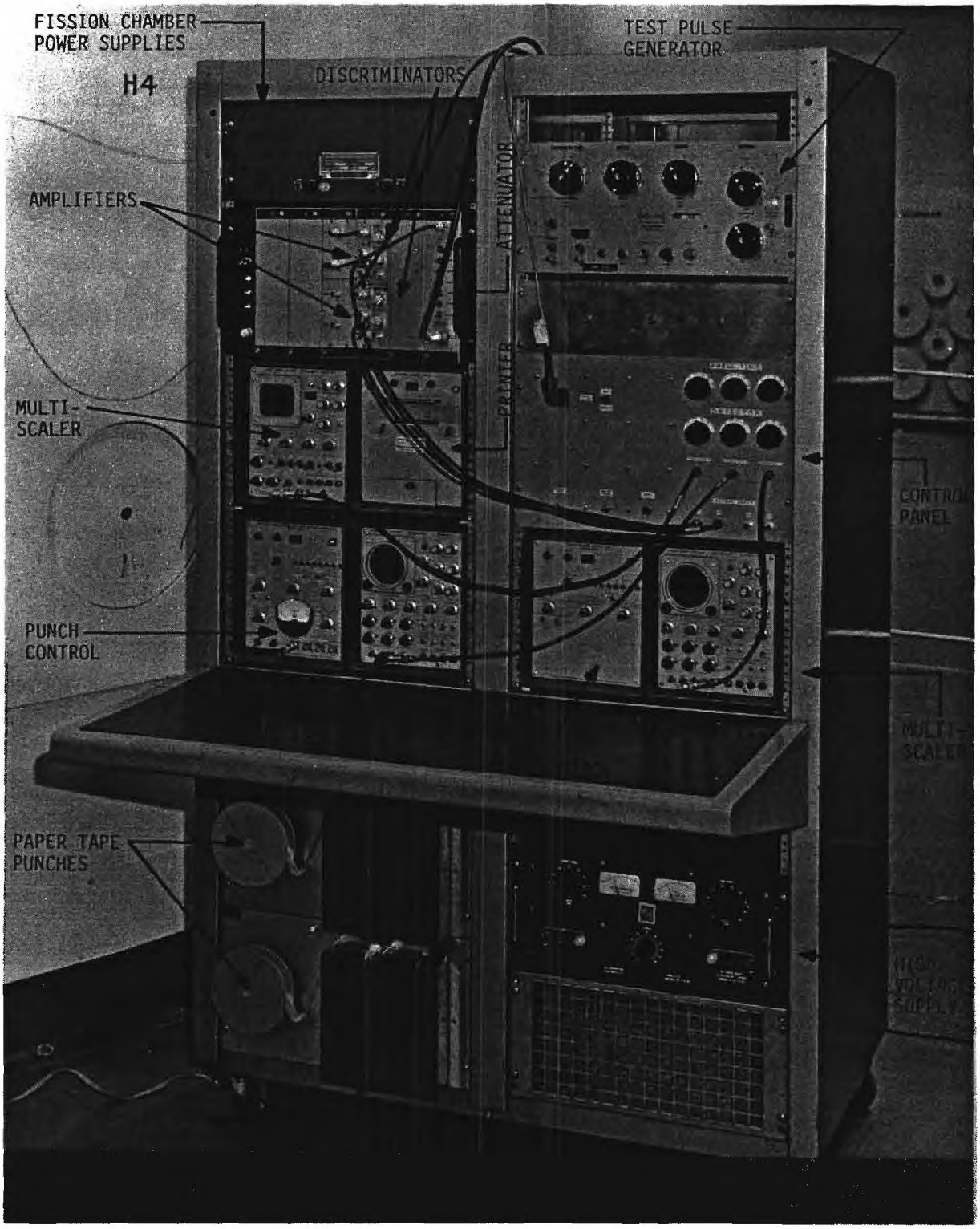


Figure 12. Detection and Data Recording Equipment at Reactor Face

CHAPTER III

PROCEDURES

Detector Positioning

The detectors described in the last chapter were positioned in instrument hole H21A for preliminary data taking and in the through-tubes H11 and H12 for the final data taking. The through-tubes were selected because they afford a position near the core from which the decaying flux may be monitored with the relatively high count rates which are desirable for good statistics. At the same time, the detector position is readily adjustable within the through-tube.

The nuclear safety requirements for the through-tubes[29] are somewhat more stringent than they are for experimental facilities which do not penetrate the core tank. Experiments, or detectors, inserted in this position must be so constructed or equipped that they will remain fixed when subjected to a three psi pressure differential, such as might occur in the case of a leak of heavy-water into the through-tube. Figures 13 and 14 illustrate the arrangement of components which provide the detector positioning and fixing requirements.

In the flux decay data reported herein, the detectors were typically positioned near their maximum count rate position. This was desirable since the reactor had been operating at low power (less than 200 watts, and usually at only one to five watts) and residual fluxes were relatively low. In future experiments, which will be run after higher power operation, it is

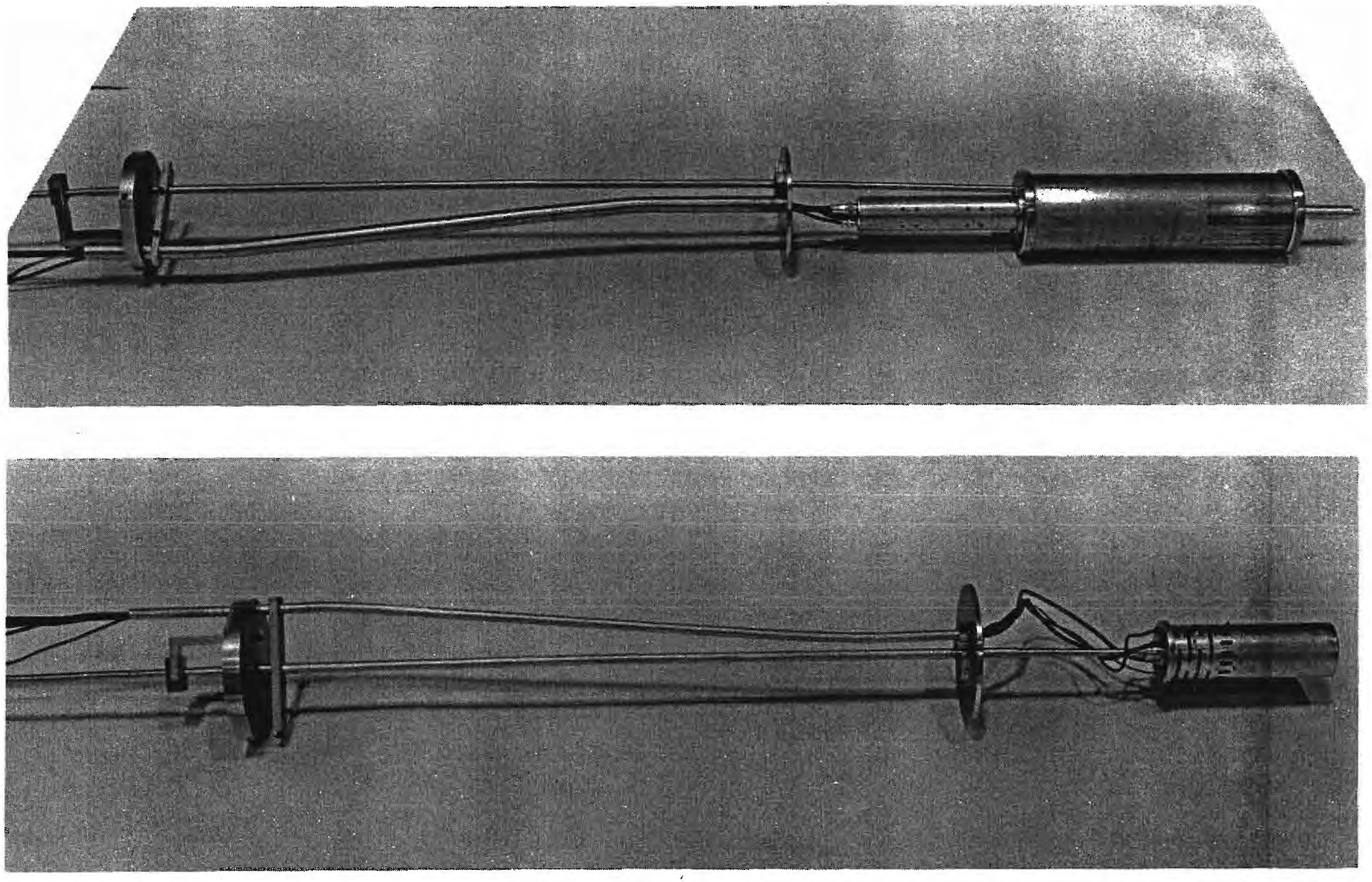


Figure 13. Detector Positioning Devices

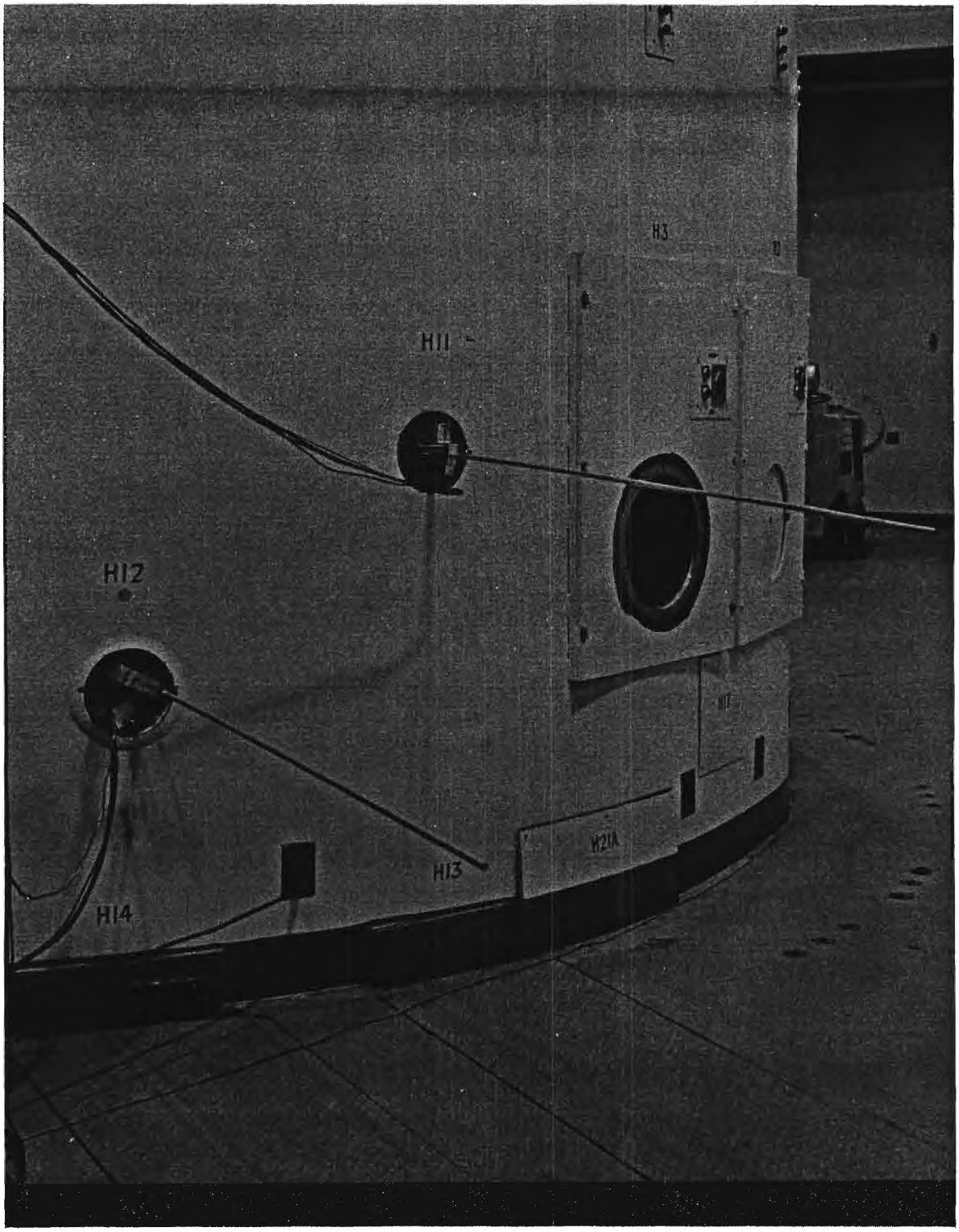


Figure 14. Detectors Positioned in Through-Tubes

expected that the adjustable feature of the detector positioning equipment will be more fully utilized.

Before the safety rods were dropped, commencing the data taking, the neutron start-up source was removed from its position in the core in order to eliminate this background contribution. Any remaining background contribution, such as spontaneous fission or photoneutrons produced by naturally occurring high energy gamma rays, appears in the flux decay analysis as a component with a negligible decay constant value.

Dead Time Determination

The dead times of the detectors employed were assessed from interpretation of the Tektronix Type 545A oscilloscope presentation of signal pulses. This procedure yielded a dead time of 400 nanoseconds for the scintillation detector and 240 nanoseconds for the fission chamber. The dead times of detector-analyzer systems were determined in an operational test in which the reactor, initially critical at low power, was put on a positive asymptotic period of about 56 seconds and the system response recorded as the flux increased. The results of this test for the fission chamber system are illustrated in Figure 15. When the reactor had settled out on an asymptotic period, the detector-analyzer system was used to record the response of the detector, which was monitoring the power rise. Counts were recorded at one second per channel, producing the count rate values represented in Figure 15. These counts were plotted. A straight line on semi-logarithmic paper was drawn through the low count rate points for which negligible counting loss was experienced. Then this line was extended to the higher count rate region. Assuming this straight line to represent the ideal or "no-lost-counts" response to reactor behavior, the comparison with recorded

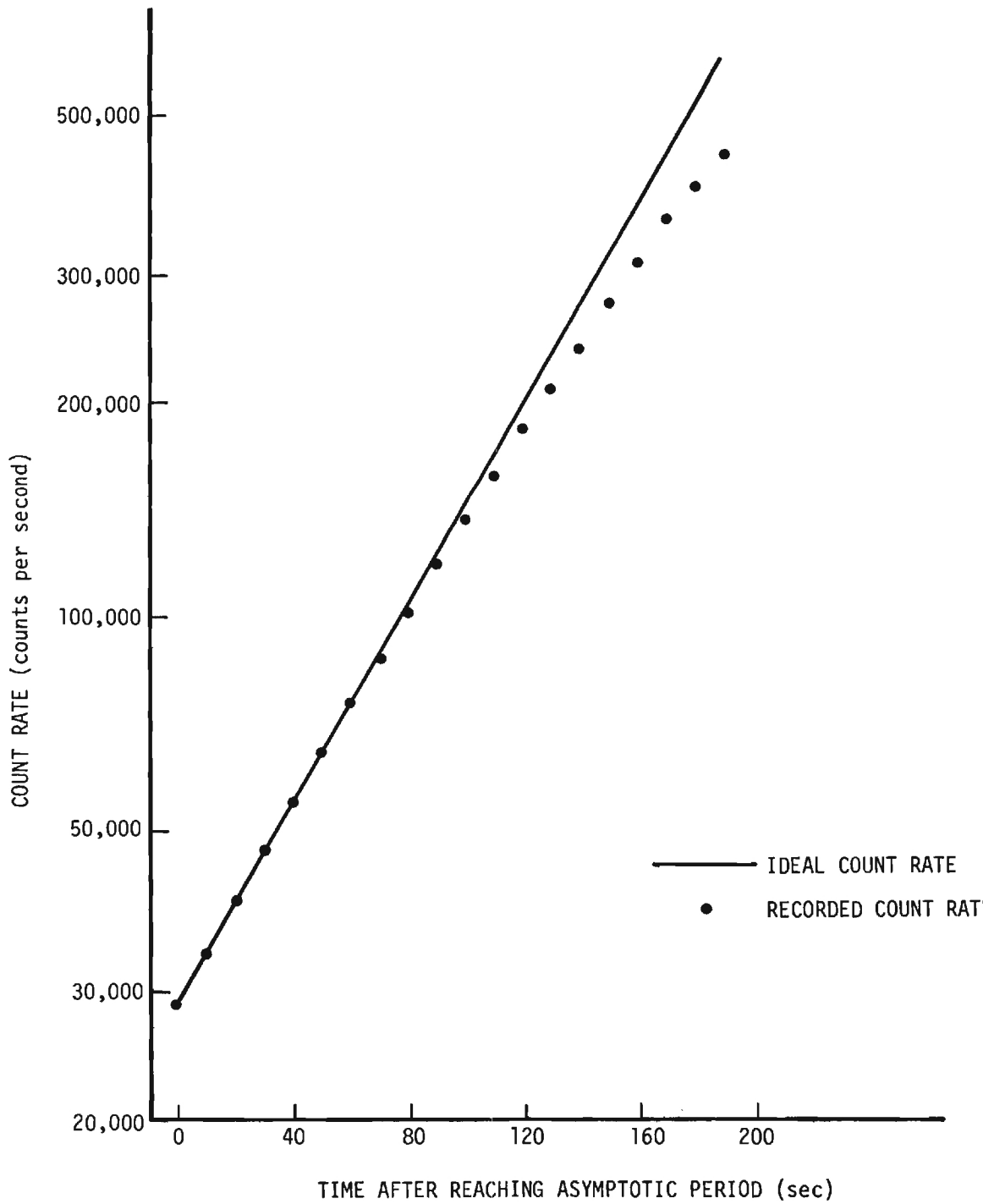


Figure 15. Detector-Analyzer System Response with Reactor on a Positive Asymptotic Period

count rates yielded a value of system resolving time. These tests indicated a total detection system pulse resolution capability of 530 nanoseconds for each of the two systems. This value corresponds with that obtained in a separate system test using a Rutherford pulse generator providing a pulse whose shape was made to resemble that of the discriminator output pulse in rise time, amplitude, and width. In this test, it was found that the maximum count rate, as observed on the oscilloscope, was also obtained for pulses separated by 530 nanoseconds.

Delayed Neutron Precursor Concentration Buildup

By referring to Figure 16, a logical flow chart of information in process, one may observe the central importance of the delayed neutron precursor concentrations. The buildup of precursors may affect the results of control rod calibrations, experiment zero time adjustment, and the correction of parameters for sub-critical multiplication. A simple assessment of the fraction of equilibrium concentration of a precursor which exists at the time a control rod is dropped to start a flux decay experiment may be made by use of the familiar formula:

$$Q_i = 1 - e^{-\lambda_i (t_p - t_d)}$$

where Q_i represents the equilibrium fraction of the i^{th} precursor, λ_i is that precursor's decay constant, and $(t_p - t_d)$ is the time interval before rod drop over which the reactor has been operating at a level power.

When the longer-lived photoneutron precursors are involved, such a procedure may lead to gross inaccuracies since the level power operation

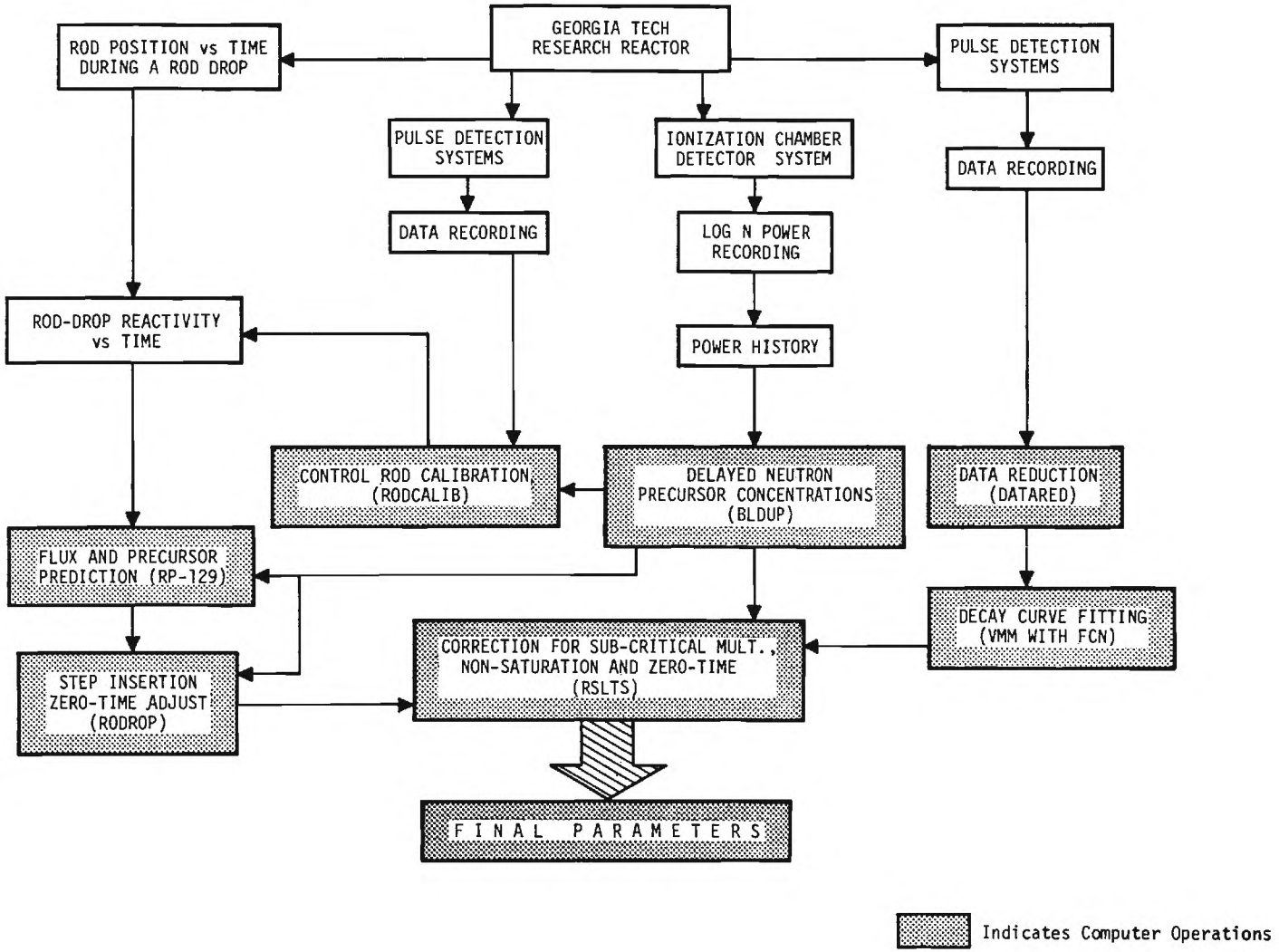


Figure 16. Logical Flow Chart of Information in Process

time may be short compared with precursor half-life. Therefore, it was decided to improve the estimate of equilibrium fraction by processing the reactor's power history using a computer program, BLDUP, developed for the Burroughs B-5500 computer at Georgia Tech. This program and the equations which it solves are included in Appendix B.

The incorporation of reactor power history into computer-readable form, i.e. punched cards, is expected to produce desirable long-term research benefits. It will permit realistic future interpretation of the residual "sources" within the reactor and will also afford the basic information necessary for an accurate burn-up assessment. It is believed that this is the first time that this refinement in calculating precursor concentrations has been included in either parameter determinations or kinetics calculations.

Rod Drop Zero Time Determination

In this research, reactor kinetics parameters have been determined by fitting the expression, $\frac{n(t)}{n(0)} = \sum_{i=1}^{M} a_i e^{-\lambda_i' t}$, to the flux decay data. This method employs the commonly used assumption that, at time zero, an instantaneous step insertion of negative reactivity was provided. The time required to drop a rod in the Georgia Tech Research Reactor has been measured by electronic and high speed photographic methods and, as one might expect, it differs considerably between rods, depending on the initial rod position before the drop and the operating characteristics of the dashpot. In any event, while more than 90 percent of the reactivity is invariably inserted in less than one-half second, this is not a truly instantaneous insertion.

It should be evident that a discrepancy of the order of one-half second is of little importance at times in the order of hours, but it may have some bearing on the results obtained for precursors having half-lives of the order of some hundred milliseconds. One might choose the start of rod drop motion as zero time, the completion of rod drop as zero time, or select some time between these two. The latter technique has been followed. In determining the constructive step insertion time, it is desired to choose a time such that a flux versus time calculation using the actual functional description of reactivity insertion versus time will produce results which match those obtained using the step description of reactivity insertion at some time after the drop.

A typical control rod has been calibrated from the banked-critical position (19.5° withdrawn to full-in) using the RODCALIB method.[30] The results are shown in Figure 17. Thus, the relation between reactivity and rod position may be estimated. Electronic measurement using a Baldwin shaft-position encoder has provided the information on rod position as a function of time in a drop. These results appear in Figure 18. Using these results, a mean value for reactivity insertion as a function of time has been plotted (Figure 19). The representation of this reactivity change as a ramp insertion completed in one-fourth second permits one to apply the ALGOL version of the computer program RP-129.[31] This formulation, originally coded at the Argonne National Laboratory in FORTRAN, solves the one-group, space-independent reactor kinetics equations which were discussed in Chapter I. The output of RP-129 consists of neutron flux and delayed neutron precursor concentrations at uniformly spaced time intervals. These results form the basic input for the ALGOL version of another ANL-originated

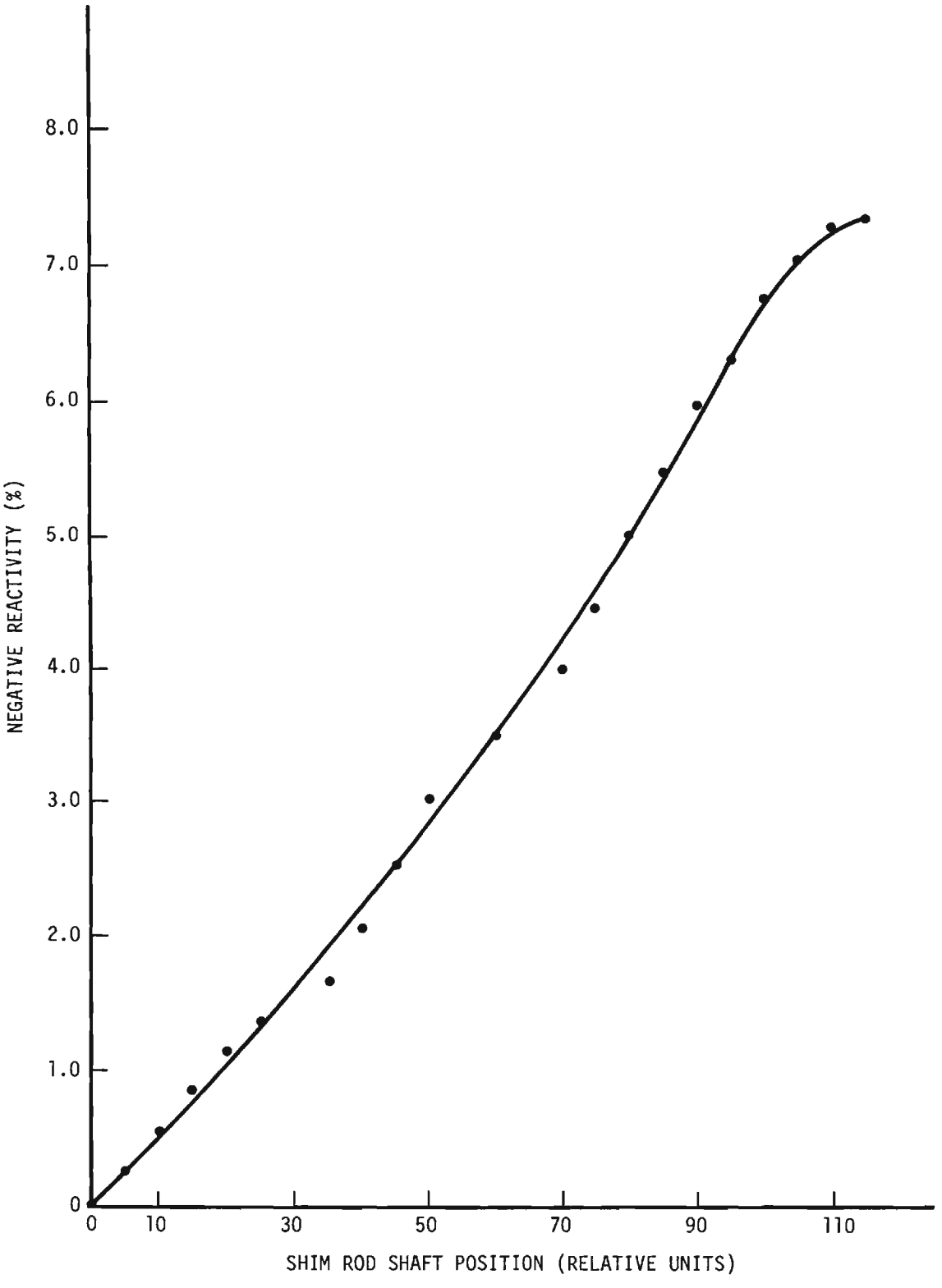


Figure 17. Shim Safety Control Rod Calibration from the Banked-Critical Position to Full-in

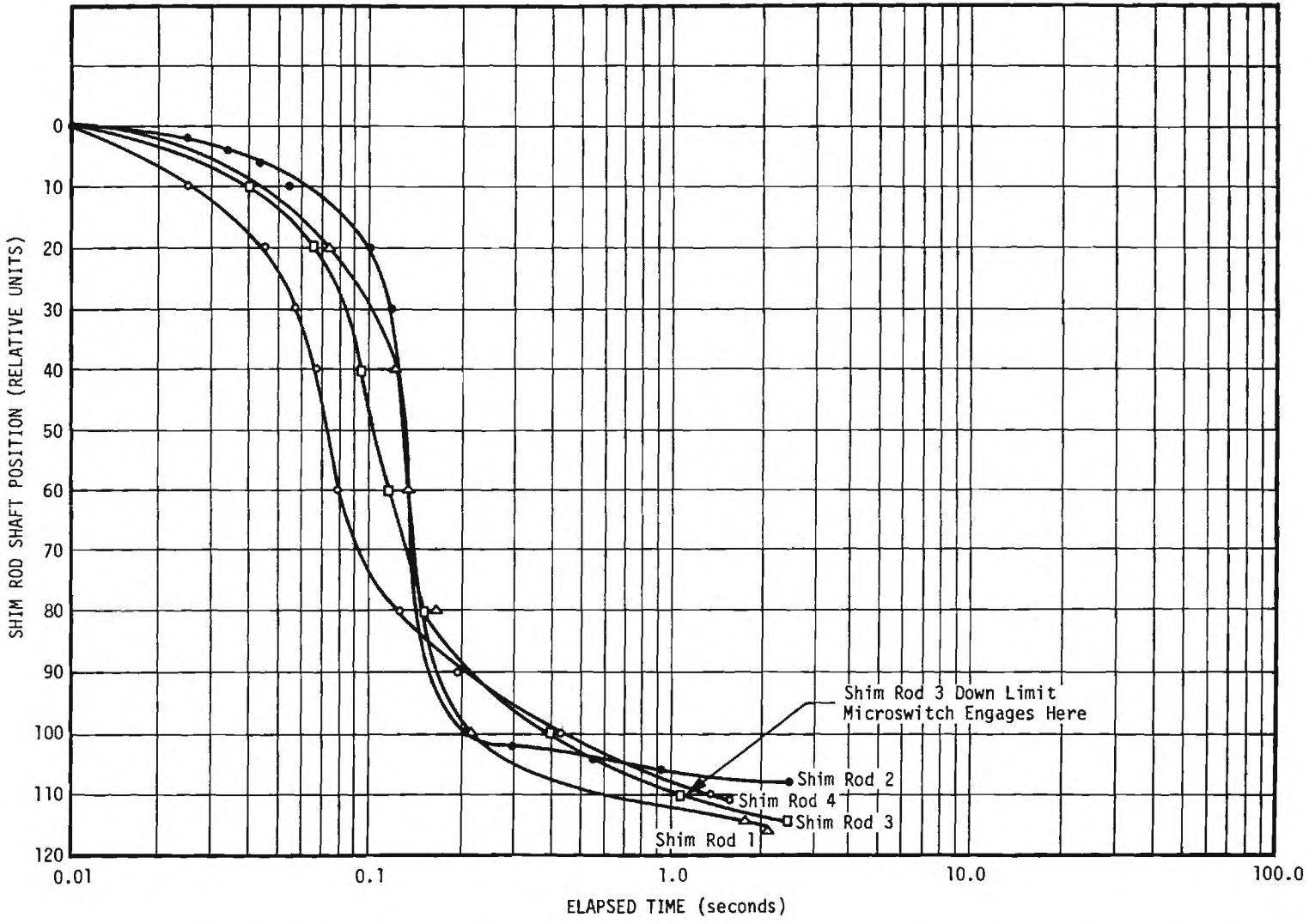


Figure 18. Shim Safety Control Rod Position as a Function of Time Following a Drop from the Banked-Critical Position

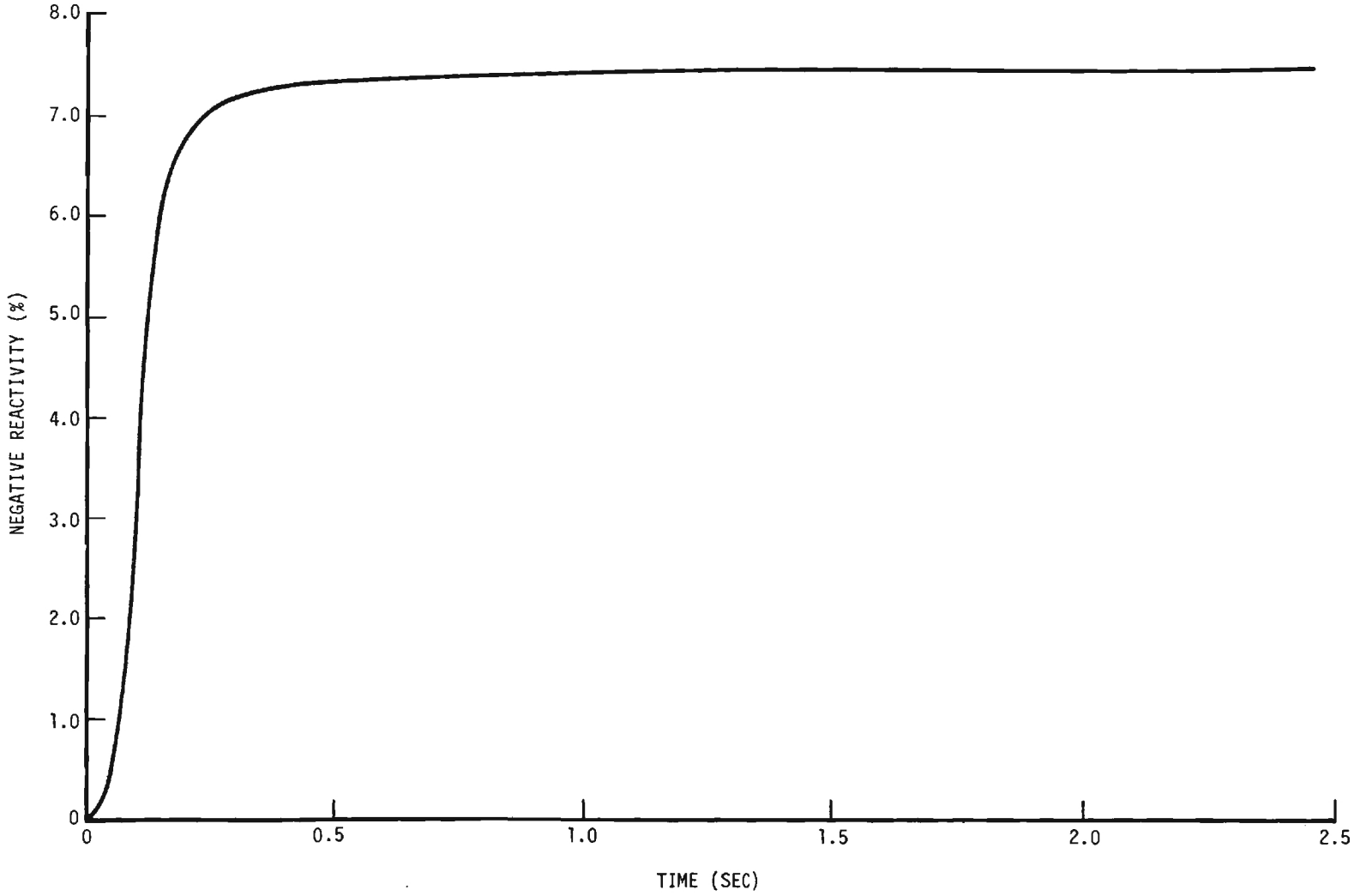


Figure 19. Shim Safety Control Rod -- Negative Reactivity Insertion as a Function of Time Following a Drop from the Banked-Critical Position

FORTRAN program entitled RODROP.[32] Both of these computer programs are included in Appendix C. The purpose of RODROP is to determine the time at which an instantaneous insertion of negative reactivity would have produced, several minutes later, a flux value which approximates closely that flux predicted by the RP-129 calculation. The RP-129 calculation was based on a more realistic ramp insertion of reactivity. Both calculations assume the same total negative reactivity insertion. They were both made using the best available estimates of kinetics parameters.[11] While one might question the desirability of using estimated kinetics parameters to obtain corrections to other kinetics parameters, this is in fact a valid procedure since this calculation is not sensitive to the details of the assumed parameters. This is particularly true because of the shape of the reactivity-versus-time curve of the rod drop in which most of the reactivity insertion occurs in a short time.

Data Collection

The data recording equipment has been described in Chapter II. The procedure followed for data collection using this equipment will now be detailed.

The fission chamber detector and scintillation detector were positioned in the through-tubes. It was considered desirable to obtain data with two different detectors in order to determine any inconsistency or detector-analyzer system malfunction. The comparative results will be discussed in Chapter IV. The reactor was started up and operated at steady power for various lengths of time. The antimony-beryllium start-up source was removed. All four shim rods were dropped and data collection initiated

on a start signal from the down-limit microswitch of one of the control rods. All three multiscalers were sequenced by pulses from the same clock. Analyzer 1 received fission chamber system counts, analyzer 2 scintillation detector system counts, and analyzer 3 received a 1 kc time pulse as a check on clock accuracy. The initial dwell time per channel setting was 0.1 sec. At preselected times, the dwell time setting was advanced. In this fashion, a single 400 channel multiscaler was capable of collecting data for several days without relief or recycle of analyzers. During the course of a counting experiment, the reactor top reflector remained at its operating level. No modifications of reactor configuration were permitted, i.e. no insertions or removals.

The punched paper tapes produced in the readout operation were converted to the punched cards required by the B-5500 computer using an I.B.M. Tape-to-Card Printing Punch with a wired program that checks for punch errors and reproduces the digitized description of the experiment on each card.

Data Reduction and Fitting

The flux decay data on punched cards contained the raw information necessary for the curve fitting Variable Metric Minimization (VMM) program,[33] but preliminary reduction and processing were required. These were accomplished by the computer procedure DATARED which is reproduced and discussed in Appendix D. The purpose of DATARED was to test the data for proper sequencing, convert multiscaler dwell time and channel number to elapsed experiment real time, correct data for detector dead time, combine original data points, if required, to provide a reasonable number of final data points

for curve fitting, and to calculate the statistical deviation of each final data point.

Weighted least squares fitting of the flux decay data to the expression $\frac{n(t)}{n(0)} = \sum_i a_i e^{-\lambda_i' t}$ was accomplished using VMM with an integral function procedure. VMM is a relatively new method for numerically determining the local minimum of a differentiable function of several variables. Instead of using matrix inversion as is frequently done in such problems, VMM starts with an estimate of the inverted matrix and attempts to improve the estimate with each iteration. This technique was devised by its originator, W. C. Davidon, in order to avoid the difficulties of ill-conditioned matrices of the sort encountered in some radioactive decay problems. A discussion of the method has been given by Davidon[33] and by Fletcher and Powell.[34] Since VMM is a general purpose method, the value of the specific function to be minimized, along with its derivatives with respect to each variable, must be calculated in a separate procedure supplied by the user. The function incorporates the goodness-of-fit criterion desired, which in this case is weighted least squares. The details of this function and its derivatives, along with the computer program itself may be found in Appendix E.

After the data have been analyzed to find the parameters producing a best fit, they must be corrected in order to make them applicable under other power history and shut-down conditions. These corrections, which are accomplished in Procedure RSLTS, account for any variation of delayed neutron precursor concentration from an equilibrium value (using BLDUP), the effect of sub-critical neutron multiplication on decay constant parameters, and the change in relative group abundance between the rod drop calculated

zero time and the time at which data collection was started (with zero time having been computed using RP-129 and RODROP). The theory by which these corrections are calculated is detailed in Appendix F.

CHAPTER IV

RESULTS

Flux decay data for up to 75 hours after shutdown were obtained on ten separate occasions during the low-power operation of the Georgia Tech Research Reactor. Several preliminary data sets were deemed unacceptable due to interval timer malfunctions or control error. On two occasions, power failures interrupted the experiment for extended periods. In one instance, at the last crucial moment the flexing of a power cable broke a solder joint, incapacitating one of the two detectors from which parallel data were desired. Undoubtedly the most elusive difficulty was that of noise interference. Until the most extreme and sometimes redundant measures for eliminating line and radiated interference were employed, random noise counts from some known and other still unknown sources were apparent. Considerable attention was given to component grounding and shielding and cable shielding. The laboratory regulated power supply was filtered by a Sorensen A.C. regulator. Special D.C. filament supplies were built for the detector preamplifiers.

Two satisfactory experiments, henceforth designated A and B, were carried out in late April and early May, 1965. In each of these, both detectors operated in an acceptable manner. In the discussion which follows, the Test A data obtained with the scintillation detector system will be referred to as A(s); that obtained with the fission chamber system will be designated A(f). Similar designations will apply to Test B results, i.e.

B(s) for scintillation detector data and B(f) for fission chamber data.

Preliminary fitting attempts very early indicated the necessity for accurate first estimates of the parameters and the uncertainty in these estimates. The square of this uncertainty estimate is a starting value for the error matrix which is refined by the Variable Metric Minimization method. This error matrix controls the magnitude of the parameter-correction step and, after fitting, provides an estimate of the precision with which each parameter has been determined. Initial estimates were based on the decay constants and relative abundances found in the literature, specifically Keepin's compilation.[11] It was found to be helpful to improve these estimates initially by treating the data collected over various time intervals as count rates associated with the mid-point of the time interval. The data collection time interval varied from 0.1 second per data point at the rod drop to 1100 seconds per data point in the final phase of counting. It is believed that this approach provides faster initial convergence because of the greater simplicity of the count rate fitting function and its derivatives.

In the first count rate function fits of the data sets A(s), A(f), B(s), and B(f), it was evident that A(s) would be amenable to the closest fit. With 349 data points, an eight-group fit resulted in an F value of 344.8 (see Equation 6 on page 123) and a weighted variance of fit, V, of 1.035. The other three tests had eight-group weighted variances of 1.21, 1.25, and 1.24. Since each fit to a different number of groups required at least one-half hour of B-5500 computer time, it was decided to select the number of groups required for the fit by investigating the A(s) data set in detail, using the exact integral fit method, and then analyzing the other

data sets with selected numbers of groups based on the A(s) outcome. Figure 20 illustrates the result of integral function fits as A(s) data were fitted to the expression,

$$n(t_j \rightarrow t_j + \Delta t_j) = \sum_{i=1}^{M} \int_{t_j}^{t_j+\Delta t_j} a_i' e^{-\lambda_i' t} \, dt$$

with a successively larger number of delayed neutron groups, M. A minimum in F value as well as figure-of-merit, V, occurred at 13 groups. One of these 13 was identified as a background term since its decay constant was 2.73×10^{-10} sec^{-1}, corresponding to a steady term. Therefore, it was concluded that these data indicate a best fit with twelve delayed neutron groups plus a background group. It should be mentioned at this point that a figure-of-merit value of about 1.0 is the region of acceptable fitting results. In this region of acceptability, roughly two-thirds of the data points will fall within one standard deviation of their calculated values and the remaining points will be sufficiently well-behaved that the average residual (difference between observed and calculated values divided by the standard deviation of the observed data) will have a value slightly less than 1. On this basis, it may be seen that almost all of the fits indicated in Figure 20 were acceptable, although the 13-group fit was best. Plots of data, calculated fit, and residuals for Tests A(s) and B(f) appear in Appendix A. The relative abundances and decay constants for the A(s) fit are presented in Table 1, along with some interesting correction factors which are worthy of comment.

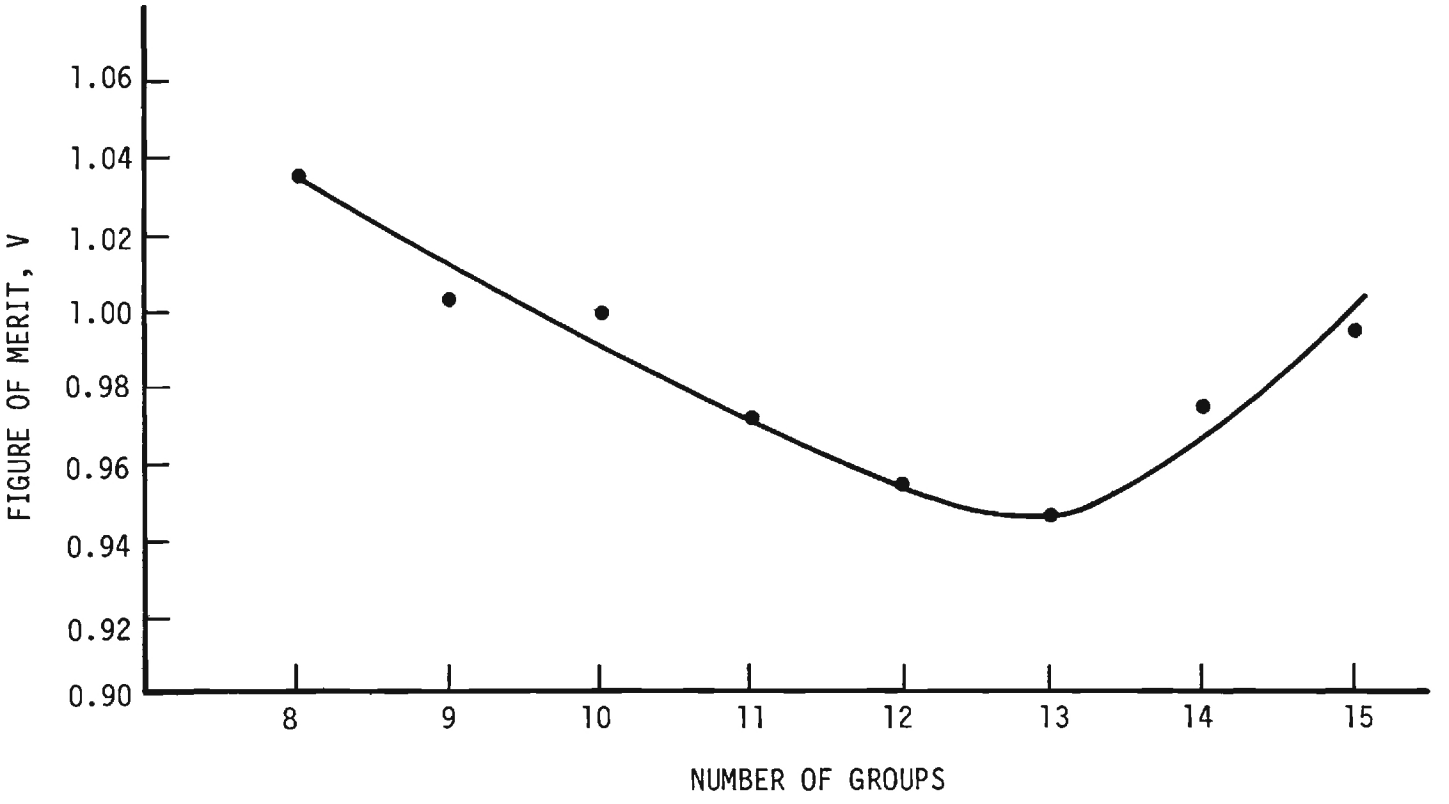


Figure 20. Figure-of-Merit for Test A(s) Parameters as a Function of Number of Delayed Neutron Groups Used in the Fitting

Table 1. Test A(s) 13-Group Parameters with Correction Terms

Group	Resolved $\lambda(sec^{-1})$	Corrected $\lambda(sec^{-1})$	Equilibrium Fraction	Zero-Time Enhancement	Relative Abundance
1	1.92	1.96(± 0.25)	1.00	8.30	0.245(± 0.043)
2	4.39×10^{-1}	$4.41(\pm 0.38) \times 10^{-1}$	1.00	1.62	0.0812(± 0.0105)
3	2.83×10^{-1}	$2.88(\pm 0.09) \times 10^{-1}$	1.00	1.36	0.273(± 0.020)
4	1.16×10^{-1}	$1.17(\pm 0.04) \times 10^{-1}$	1.00	1.14	0.160(± 0.015)
5	3.01×10^{-2}	$3.05(\pm 0.05) \times 10^{-2}$	1.00	1.03	0.198(± 0.004)
6	9.22×10^{-3}	$9.25(\pm 0.28) \times 10^{-3}$	0.988	1.01	0.0337(± 0.0022)
7	2.27×10^{-3}	$2.27(\pm 0.32) \times 10^{-3}$	1.37	1.00	0.00309(± 0.00026)
8	7.73×10^{-4}	$7.73(\pm 0.14) \times 10^{-4}$	5.94	1.00	0.00200(± 0.00008)
9	3.15×10^{-4}	$3.15(\pm 0.06) \times 10^{-4}$	7.55	1.00	0.00142(± 0.00004)
10	1.18×10^{-4}	$1.18(\pm 0.004) \times 10^{-4}$	4.82	1.00	0.00225(± 0.00003)
11	5.85×10^{-5}	$5.85(\pm 0.06) \times 10^{-5}$	2.80	1.00	0.000503(± 0.000014)
12	1.37×10^{-5}	$1.37(\pm 0.07) \times 10^{-5}$	0.993	1.00	0.0000272(± 0.0000014)
13 (Background)	2.73×10^{-10}	$2.73(\pm 0.08) \times 10^{-10}$	-----	----	0.58 counts/sec

Figure 21 is a reproduction of the log n power record for the GTRR at-power periods immediately preceding the initiation of Test A by rod drop. It reveals an approximate half-hour interval at 20 watts power followed by a shutdown (to insert detectors) and a return to 1 watt power for about 10 minutes before rod drop. Since it was planned to account for precursor equilibrium fraction by an analysis of the power history, as discussed in Appendix B, and since the reactor had not operated at appreciable power since startup, this mode of operation was chosen in order to enhance the contribution to the decaying flux of intermediate and longer-lived precursors relative to the short-lived components whose presence would be readily apparent in any event. The result of this enhancement is evident in the equilibrium fraction column of Table 1, in which it may be seen that some groups even exceeded their saturation value, based on power level at the drop. This was possible since precursor atoms generated at the 20 watt level had not had sufficient time to decay away and reach equilibrium at the 1 watt level. The group possessing the largest equilibrium fraction, 7.55, has a half-life of 36.6 minutes and was thus short-lived enough to be at nearly 50 percent equilibrium based on the 20 watt level and sufficiently long-lived not to decay away markedly before shutdown. This equilibrium fraction provided a correction term that was applied before the calculation of the relative abundances in order that all abundances might be based on the saturation value of precursors.

The second correction factor in Table 1, labeled zero-time enhancement, is due to the 1.1 second lag between the time of the constructive step insertion of reactivity and the time that the down-limit microswitch on shim rod #3 was engaged, starting data collection. The constructive zero-

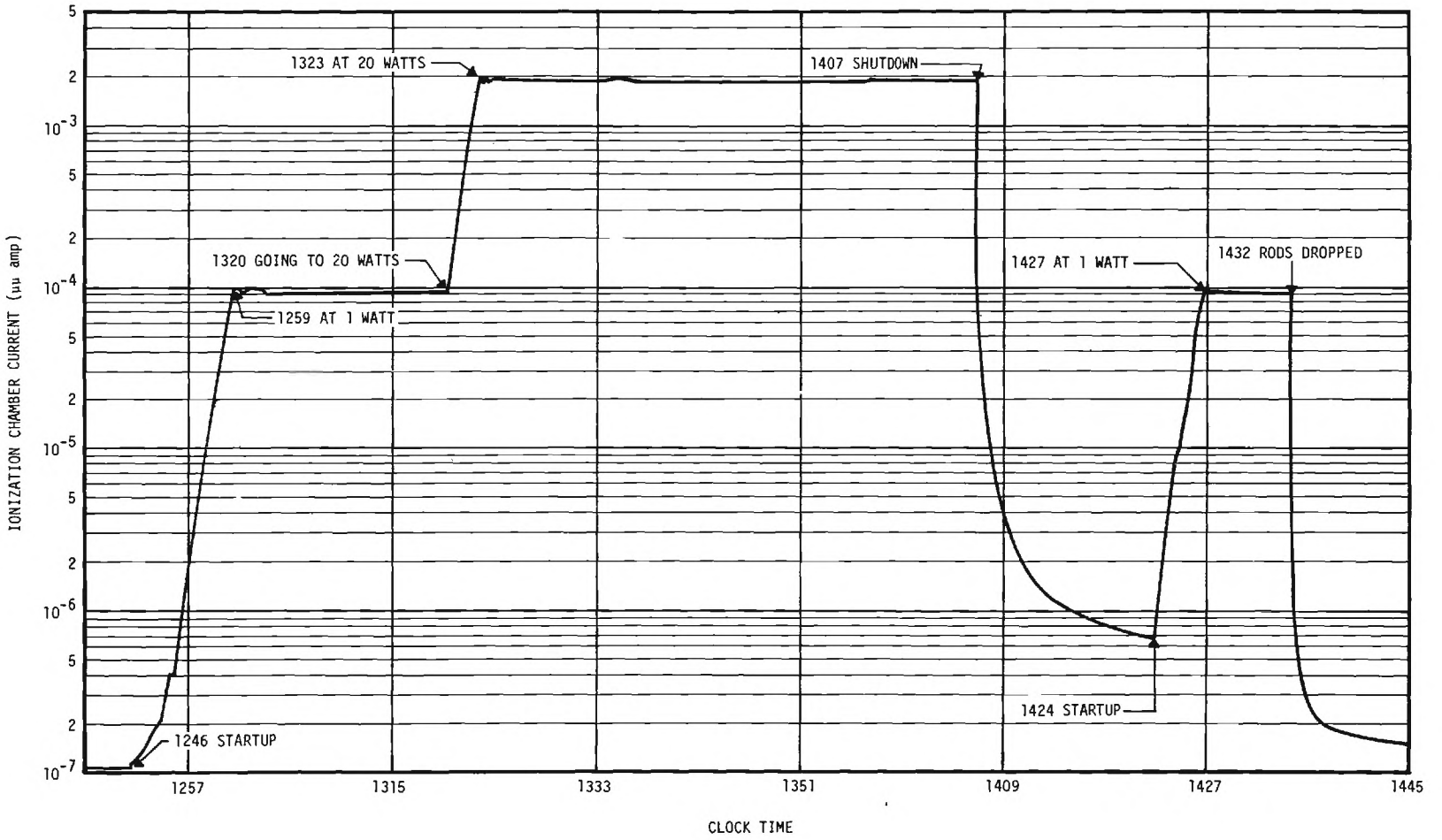


Figure 21. Log n Power Recording for the Georgia Tech Research Reactor At-Power Periods Immediately Preceding the Start of Test A

time for these experiments was calculated by the methods described in Appendix C to be 0.038 second, and the total drop time from the banked-critical position to the engaging of the microswitch was measured and found to be 1.13 to 1.14 seconds. This enhancement was most apparent in the shortest-lived group with a decay constant of 1.96 ± 0.25 sec^{-1}. This corresponds to a half-life of 0.353 second; thus over a 1.1 second interval this group would be expected to decay away to less than one-eighth of its zero-time abundance.

Table 2 presents the parameters obtained by fitting the data of Test A(s) with several different numbers of delayed neutron groups. In order to ascertain how good a fit Keepin's suggested decay constants would provide for these data, we have included a fit to 14 groups, in which the abundances were free to be varied by the VMM program while the decay constants were constrained to remain at the values suggested by Keepin,[11] with one exception. The shortest-lived group, with a half-life of 0.179 second, was omitted since the conditions of this experiment did not favor its resolution, if it does in fact exist. The resolution of this group was sought in a special rod drop experiment with the GTRR in which data were collected at 0.1 second per channel from before a drop until 22 seconds after the drop and analyzed beginning at 0.5 second after the start of rod motion. The best fit to the data of this special experiment resulted when the shortest-lived group had a half-life of 0.452 second, indicating that a 0.179 second group could not be resolved.

Table 3 compares the 13-group fits of Tests A(s), A(f), B(s), and B(f). Test B(s) did not fit well and the 13-group fit to its data resulted in a weighted variance of 1.38. Since this figure-of-merit indicates a

Table 2. Test A(s) Parameters

Groups	8 Group Fit V = 1.035		10 Group Fit V = 1.005		12 Group Fit V = 0.9568		13 Group Fit V = 0.9470		14 Group Fit with Literature[11] Values of λ_i V = 1.215	
	$\beta_i$eff/βeff	λ_i(sec^{-1})	$\beta_i$eff/βeff	λ_i(sec^{-1})	$\beta_i$eff/βeff	λ_i(sec^{-1})	$\beta_i$eff/βeff	λ_i(sec^{-1})	$\beta_i$eff/βeff	λ_i(sec^{-1})
1					0.174	1.63	0.245	1.96	0.0645	1.40
2					0.311	3.27×10^{-1}	0.0812	4.41×10^{-1}	0.332	3.11×10^{-1}
3	0.628	2.37×10^{-1}	0.606	2.57×10^{-1}	0.0893	2.62×10^{-1}	0.273	2.88×10^{-1}	0.0992	2.77×10^{-1}
4					0.176	1.04×10^{-1}	0.160	1.17×10^{-1}	0.206	1.15×10^{-1}
5	0.312	3.33×10^{-2}	0.231	4.32×10^{-2}	0.212	2.82×10^{-2}	0.198	3.05×10^{-2}	0.214	3.17×10^{-2}
6			0.0972	2.46×10^{-2}					0.0264	1.69×10^{-2}
7	0.0489	8.33×10^{-3}	0.0538	9.77×10^{-3}	0.00660	8.97×10^{-3}	0.0337	9.25×10^{-3}	0.0375	1.27×10^{-2}
8					0.0232	6.83×10^{-3}	0.00309	2.27×10^{-3}	0.00982	4.81×10^{-3}
9	0.00348	8.89×10^{-4}	0.00376	1.06×10^{-3}	0.00268	8.84×10^{-4}	0.00200	7.73×10^{-4}	0.00370	1.50×10^{-3}
10	0.00222	3.03×10^{-4}	0.00232	3.54×10^{-4}	0.00174	3.15×10^{-4}	0.00142	3.15×10^{-4}	0.00278	4.28×10^{-4}
11	0.00320	1.09×10^{-4}	0.00319	1.18×10^{-4}	0.00258	1.13×10^{-4}	0.00225	1.18×10^{-4}	0.00324	1.17×10^{-4}
12	0.000398	4.54×10^{-5}	0.000613	5.39×10^{-5}	0.000401	4.97×10^{-5}	0.000503	5.85×10^{-5}	0.000467	4.37×10^{-5}
13			0.0000204	3.65×10^{-6}			0.0000272	1.37×10^{-5}	0.0000212	3.63×10^{-6}
14	0.00151	5.94×10^{-7}	0.00134	5.94×10^{-7}	0.000134	7.77×10^{-7}			0.000305	6.24×10^{-7}
Back-ground							0.58 counts per second	2.7×10^{-10}		

Table 3. Comparison of 13-Group Parameters - Tests A and B

Group	Test A(s) V = 0.947		Test A(f) V = 1.14		Test B(f) V = 1.07		Test B(s) V = 1.38	
	$\beta_i$eff/βeff	λ_i(sec^{-1})	$\beta_i$eff/βeff	λ_i(sec^{-1})	$\beta_i$eff/βeff	λ_i(sec^{-1})	$\beta_i$eff/βeff	λ_i(sec^{-1})
1	0.245(±0.043)	1.96 (±0.25)	0.286(±0.028)	1.99 (±0.22)	0.175(±0.018)	1.61 (±0.16)	0.518(±0.52)	2.07(±0.20)
2	0.0812(±0.0105)	4.41 (±0.38) $\times 10^{-1}$	0.260(±0.017)	3.56 (±0.20) $\times 10^{-1}$	0.297(±0.018)	3.76 (±0.23) $\times 10^{-1}$	0.0207(±0.0021)	7.43(±0.74) $\times 10^{-1}$
3	0.273(±0.020)	2.88 (±0.09) $\times 10^{-1}$	0.0764(±0.0054)	2.35 (±0.12) $\times 10^{-1}$	0.0807(±0.0084)	2.38 (±0.21) $\times 10^{-1}$	0.230(±0.023)	2.19(±0.21) $\times 10^{-1}$
4	0.160(±0.015)	1.17 (±0.04) $\times 10^{-1}$	0.154(±0.011)	1.16 (±0.04) $\times 10^{-1}$	0.205(±0.012)	1.12 (±0.05) $\times 10^{-1}$	0.0497(±0.0050)	1.79(±0.18) $\times 10^{-1}$
5	0.198(±0.004)	3.05 (±0.05) $\times 10^{-2}$	0.185(±0.002)	3.04 (±0.04) $\times 10^{-2}$	0.209(±0.003)	2.76 (±0.03) $\times 10^{-2}$	0.155(±0.015)	3.26(±0.27) $\times 10^{-2}$
6	0.0337(±0.0022)	9.25 (±0.28) $\times 10^{-3}$	0.0115(±0.0009)	1.26 (±0.06) $\times 10^{-2}$	0.0209(±0.0012)	8.73 (±0.34) $\times 10^{-3}$	0.00664(±0.00066)	8.19(±0.81) $\times 10^{-3}$
7	0.00309(±0.00026)	2.27 (±0.32) $\times 10^{-3}$	0.00206(±0.0013)	7.32 (±0.27) $\times 10^{-3}$	0.00685(±0.00055)	4.02 (±0.14) $\times 10^{-3}$	0.0160(±0.0016)	7.41(±0.68) $\times 10^{-3}$
8	0.00200(±0.00008)	7.73 (±0.14) $\times 10^{-4}$	0.00217(±0.00004)	8.56 (±0.09) $\times 10^{-4}$	0.00204(±0.00006)	9.21 (±0.14) $\times 10^{-4}$	0.00167(±0.00017)	6.54(±0.65) $\times 10^{-4}$
9	0.00142(±0.00004)	3.15 (±0.06) $\times 10^{-4}$	0.00122(±0.00002)	3.17 (±0.05) $\times 10^{-4}$	0.00133(±0.00002)	3.45 (±0.06) $\times 10^{-4}$	0.000119(±0.000012)	2.30(±0.23) $\times 10^{-4}$
10	0.00225(±0.00003)	1.18 (±0.004) $\times 10^{-4}$	0.00204(±0.00001)	1.195(±0.006) $\times 10^{-4}$	0.00175(±0.00002)	1.239(±0.009) $\times 10^{-4}$	0.00102(±0.00009)	1.66(±0.11) $\times 10^{-4}$
11	0.000503(±0.000014)	5.85 (±0.06) $\times 10^{-5}$	0.000464(±0.000011)	5.78 (±0.03) $\times 10^{-5}$	0.000412(±0.000012)	6.12 (±0.04) $\times 10^{-5}$	0.000662(±0.000059)	8.11(±0.57) $\times 10^{-5}$
12	0.0000272(±0.0000014)	1.37 (±0.07) $\times 10^{-5}$	0.0000207(±0.0000012)	1.10 (±0.08) $\times 10^{-5}$	0.0000128(±0.0000007)	1.07 (±0.04) $\times 10^{-5}$	0.0000227(±0.0000023)	2.27(±0.23) $\times 10^{-5}$
Back-ground	0.58 counts/sec	2.73 (±0.08) $\times 10^{-10}$	0.52 counts/sec	8.57 (±0.67) $\times 10^{-11}$	0.70 counts/sec	1.39 (±0.11) $\times 10^{-10}$	0.52 counts/sec	2.48(±0.25) $\times 10^{-10}$

poorer fit than that obtained in Tests A(s), A(f), and B(f), these parameters are included only to demonstrate that, even in this worst case, the results are still comparable to those obtained in relatively more successful fits. No conclusion has been reached as to why these data were less successfully fitted, especially since the same detector was used in Test B(s) as was used when the best results were obtained in Test A(s). A comparison of the remaining parameters in Table 3 reveals only a fair consistency and adequate agreement only in the case of some decay constants. In Test A it appears that there may be some degree of ambiguity between groups 2 and 3, which have decay constants of a similar magnitude, but which are not resolved to within their respective error estimates in the two different data sets. In Test A(s), group 3 abundance was roughly 3 times that of group 2, while in Test A(f) group 2 had an abundance something over 3 times that of group 3. This occurrence may be indicative of the reason for some lack of agreement as to parameter values among the different tests. The nature of this fitting process is such that a certain set of data is best fitted with a number of delayed neutron group parameters that is almost certainly smaller than the actual number of delayed neutron precursors present. The extent to which real groups merge together into the resolved groups and the details of such a merger may be responsible for the occurrence of two somewhat different sets of parameters, both of which permit an acceptable fit to the data of a given experiment as gathered by two different detectors. Thus, in Test A, it appears possible that the true physical situation affecting that portion of the experiment attributable to groups 2 and 3 may be satisfactorily represented in a non-unique manner. There is also a possibility that the VMM program did not accurately assess the error

matrix for all parameters for all tests. The refinement of that error matrix is accomplished in the program by making numerous steps in a random direction away from the minimum, and then finding the minimum again. In the process of searching for the minimum, additional information is gained which is used to improve the accuracy of the error matrix. More than 15 random steps were taken for Test A(s) and approximately 10 each for Tests A(f) and B(f). Each random step required about 10 - 15 minutes of B-5500 computer time. In any such experiment one would also be obliged to mention the possibility of detector gain-shift. This potential error-producing mechanism would be most appropriately suspected with respect to the scintillation detector. It has been previously mentioned in discussing this equipment that its design was directed toward minimizing gain shift from high to low counting rates, and careful consideration was given to the stability of the high voltage supply selected for this detector.

In Table 4 we have compared our best fit parameters of Test A(s) with those recommended by Keepin[11] (these incorporate the results of Keepin et al.,[12] Bernstein et al.,[10] and Ergen[17]), and those found in ZEEP.[19] The Keepin-Bernstein-Ergen data (henceforth to be referred to as KBE) have been combined into a single set and normalized to a total relative abundance of 1. The 1947 ZEEP experiment was conducted in order to study photoneutron parameters and the abundances were normalized to that of a prominent standard delayed group resolved from the ZEEP data which coincided in decay constant with a group reported by Hughes et al.,[9] using the Hughes abundance as the standard. Therefore, no complete listing of relative abundances can be derived from that data. The decay constants from ZEEP[19] are included for historical reasons and to indicate the

Table 4. Comparison of 13-Group Parameters of the Present Work with Literature Values

Group	Test A(s)		Avg. of Tests A(s), A(f), and B(f)		Keepin-Bernstein-Ergen[11]			Johns and Sargent[19]
	$\beta_i$eff/βeff	λ_i(sec^{-1})	$\beta_i$eff/βeff	λ_i(sec^{-1})	β_i/β	Type	λ_i(sec^{-1})	λ_i(sec^{-1})
1					0.0225	Fission	3.87	
2	0.245	1.96	0.235	1.85	0.111	Fission	1.40	
3	0.0812	4.41×10^{-1}	0.213	3.91×10^{-1}	0.352	Fission	3.11×10^{-1}	
4	0.273	2.88×10^{-1}	0.143	2.54×10^{-1}	0.0867	Photo	2.77×10^{-1}	
5	0.160	1.17×10^{-1}	0.173	1.15×10^{-1}	0.163	Fission	1.15×10^{-1}	1.72×10^{-1}
6	0.198	3.05×10^{-2}	0.197	2.95×10^{-2}	0.184	Fission	3.17×10^{-2}	3.33×10^{-2}
7	0.0337	9.25×10^{-3}	0.0220	1.02×10^{-2}	0.0272	Photo	1.69×10^{-2}	
8					0.0329	Fission	1.27×10^{-2}	1.22×10^{-2}
9	0.00309	2.27×10^{-3}	0.00400	4.54×10^{-3}	0.00932	Photo	4.81×10^{-3}	3.05×10^{-3}
10	0.00200	7.73×10^{-4}	0.00207	8.50×10^{-4}	0.00448	Photo	1.50×10^{-3}	7.30×10^{-4}
11	0.00142	3.15×10^{-4}	0.00132	3.26×10^{-4}	0.00276	Photo	4.28×10^{-4}	2.30×10^{-4}
12	0.00225	1.18×10^{-4}	0.00201	1.21×10^{-4}	0.00312	Photo	1.17×10^{-4}	1.12×10^{-4}
13	0.000503	5.85×10^{-5}	0.000460	5.92×10^{-5}	0.000430	Photo	4.37×10^{-5}	4.52×10^{-5}
14	0.0000272	1.37×10^{-5}	0.0000202	1.18×10^{-5}	0.0000137	Photo	3.63×10^{-6}	
15					0.0000666	Photo	6.24×10^{-7}	
Background	0.58 cps	2.73×10^{-10}	0.60 cps	1.66×10^{-10}				

number of groups which Johns and Sargent were able to resolve in their experiment which was similar to our own.

We have given for comparison in Table 4 a combination set of parameters obtained by averaging the corresponding groups of Tests A(s), A(f), and B(f). While this is surely a plausible set for kinetics calculations in highly enriched heavy-water reactors, and we have weighed the advisability of adopting them, we choose to use the Test A(s) parameters in the remainder of this work. We feel that they are supported by partial agreement with the A(f) and B(f) results, but are a better choice than an average with these other sets would be since a more precise fit was obtained in Test A(s).

A comparison between Test A(s) and KBE parameters is instructive, even though it is not precise. The first group of Test A(s) appears to have a decay constant, 1.96 sec^{-1}, which is close to an abundance-weighted average of the first two groups of the KBE values, i.e.

$$\frac{0.02251 \times 3.87 + 0.1108 \times 1.4}{0.1333} = 1.81$$

KBE groups 3 and 4 are difficult to compare with corresponding groups from Test A(s) due to the abundance ambiguity discussed earlier. There is, however, agreement with the decay constants of KBE groups 5 and 6. Group 6 is the important 22-second half-life by which the Bernstein photoneutron data have been incorporated into the recommended group constants of Keepin.[11] (Originally Bernstein et al. reported their results with respect to the 22-second group of Hughes et al.[9]) Since both of these groups are attributed to "standard" delayed neutrons, one may make the assumption (in agreement

with calculations made by J. Lewins for the Massachusetts Institute of Technology Reactor[26]) that $\sum_i \beta_i$eff for the GTRR is about the same as $\sum_i \beta_i$ for the KBE parameters. Then the relative abundances of these two groups may be examined for some clue as to relative effectiveness of this category of delayed neutrons in the GTRR. The ratio of Test A(s) abundances to KBE abundances for these two groups is 1.03, suggesting a 3 percent enhancement of delayed neutron importance in the GTRR due to lower average energy. The only photoneutron group with notable agreement in decay constant value is group 12. The Test A(s) value is 1.18×10^{-4} sec^{-1} and the KBE value is 1.17×10^{-4} sec^{-1}. With such close agreement on decay constant, one might anticipate that the ratios of relative abundances for this group could provide an estimate of the photoneutron effectiveness in the GTRR. This ratio, 0.72, is in fact quite close to the value of 0.691 calculated for the MITR.[26]

An interesting feature appeared in the analysis of Test A(f) results. Fits had been obtained with moderate success using eight and nine groups. The ten-group fit was surprising in that one of the abundance terms was negative, a possibility permitted by the VMM program. When the decay constant, 1.96×10^{-5} sec^{-1} was converted to a half-life value, 9.82 hours, it was recognized as being near the 9.13 hour half-life of Xe-135. Xe-135 effects had been anticipated after higher power operation, but had not been expected during low-power operation. If, indeed, this were Xe-135 it could explain the negative abundance value since the decay of a neutron sink (Xe-135) has the opposite effect of the decay of a neutron source (delayed neutron emitter). But the significance of the supposed xenon contribution was small, - 1.5 counts per second in the initial count rate of about 12,000 counts per second. A decay constant of 1.86×10^{-5} sec^{-1} was found in con-

junction with a negative abundance term in the ten-group analysis of the results of Test B(s). Other than these two rather uncertain indications, such a term did not again appear until a 16-group fit to the Test A(s) data revealed a - 0.7 count per second contribution with a decay constant, 2.09×10^{-5} sec^{-1}, convertible to a half-life value of 9.21 hours. This 16-group fit also included another low-abundance term with positive sign whose decay constant was close to that of I-135. The concentration of xenon as a function of time following shutdown can, of course, be derived from a solution of the differential equations describing the change in concentration of iodine and xenon. This has been done by Meghreblian and Holmes[35] and the result is

$$Xe(t) = \left(Xe_o + \frac{\lambda_I I_o}{\lambda_I - \lambda_{Xe}} \right) e^{-\lambda_{Xe} t} - \left(\frac{\lambda_I I_o}{\lambda_I - \lambda_{Xe}} \right) e^{-\lambda_I t}$$

where Xe_o is the initial concentration at time zero of Xe-135, I_o is the time zero concentration of I-135, and the λ's are the iodine and xenon decay constants.

If xenon were present to any appreciable extent, its change in concentration would have to be interpreted as a change in reactivity and, hence, a change in the sub-critical multiplication of the reactor. The magnitude of this effect at the time these data were collected could not exceed the value appropriate for 200 watt operation. According to Glasstone and Edlund,[36] for fluxes less than 10^{11} neutrons/cm^2sec, the Xe-135 poisoning effect at shutdown, P_o, may be estimated by

$$P_o \cong 8 \times 10^{-15} \phi_o$$

where ϕ_o is the pre-shutdown at-power neutron flux, and the poisoning, P_o, is defined as the pre-shutdown ratio of the number of thermal neutrons absorbed by the poison to those absorbed in the fuel. At such flux levels, there is no appreciable poisoning increase after shutdown. If we assume ϕ_o at 1000 kW to be 2×10^{13} n/cm^{2}sec[27] and scale down linearly to 200 watts, $\phi_o = 4 \times 10^9$ n/cm^{2}sec. Thus, $P_o \cong 3.2 \times 10^{-5}$. This effect is so small that one would reasonably consider it to be negligible. The possibility that these data do actually indicate the presence of Xe-135 is therefore difficult to evaluate. It is concluded that these indications result either from a true fitting to Xe-135 decay near the threshold of significance or a chance coincidence of the fitting results.

CHAPTER V

CONCLUSIONS

Experiments have been performed with the Georgia Tech Research Reactor to determine the number of identifiable delayed neutron groups along with the parameter values for these groups. This information is required in reactor kinetics calculations.

The parameters obtained in Test A(s) of this work represent the most appropriate values of decay constants and relative abundances presently available for a highly enriched heavy-water reactor. Twelve delayed neutron and photoneutron groups are required. Since these parameters were obtained through experiments using an actual reactor of this type, they incorporate the factors affecting such values in a direct fashion. It is for this reason that the use of these parameters in kinetics calculations should represent an improvement over calculations using parameters determined by the small-sample method. The parameters reported have been given general utility by careful attention to the corrections required to account for reactor power history, finite time of rod drop, and sub-critical multiplication. It is believed that this is the first time that these methods for power history and rod drop assessment have been applied to the kinetics parameter determination problem.

In order to illustrate the value of this work and the extent to which the particular parameters one uses may affect kinetics calculations, two different sets of calculations have been made. These were first, the

determination of the reactivity calibration curve for the GTRR regulating rod, and second, the assessment of GTRR reactivity as a function of the reciprocal of positive asymptotic period. In the regulating rod problem, with the reactor initially critical, the neutron flux was recorded while the rod was driven at constant speed from full-out to full-in. These data were analyzed using the RODCALIB method,[30] first with KBE parameters and then with the parameters determined in the present work. The RODCALIB method is a fast calibration technique that uses a digital computer program to apply the reactor kinetics equations. Since the data that are required to perform a complete control rod calibration may be obtained in less than one minute, it is an attractive method, and one that will probably be employed frequently to provide up-to-date calibrations for this control element, which is itself a commonly used secondary standard by which other reactivities are determined.

The results of the two calculations are represented in Figure 22. It may be seen in this figure that the KBE parameters produce a calibration curve that is generally about 10 percent higher than that obtained with our parameters. Thus, negative reactivity calibrations made using the KBE parameter curve for the regulating rod as a standard will be overestimated in worth by 10 percent. The excess positive reactivity which can be compensated with this rod would also be overestimated to the same extent. In both of these rod calibration calculations, the same value of total β was used, i.e. $\sum_i \beta_i = \sum_i \beta_i$eff. It should be emphasized at this point that no value for total βeff has been obtained in this work; only decay constants and relative abundances for the delayed neutron groups have been measured.

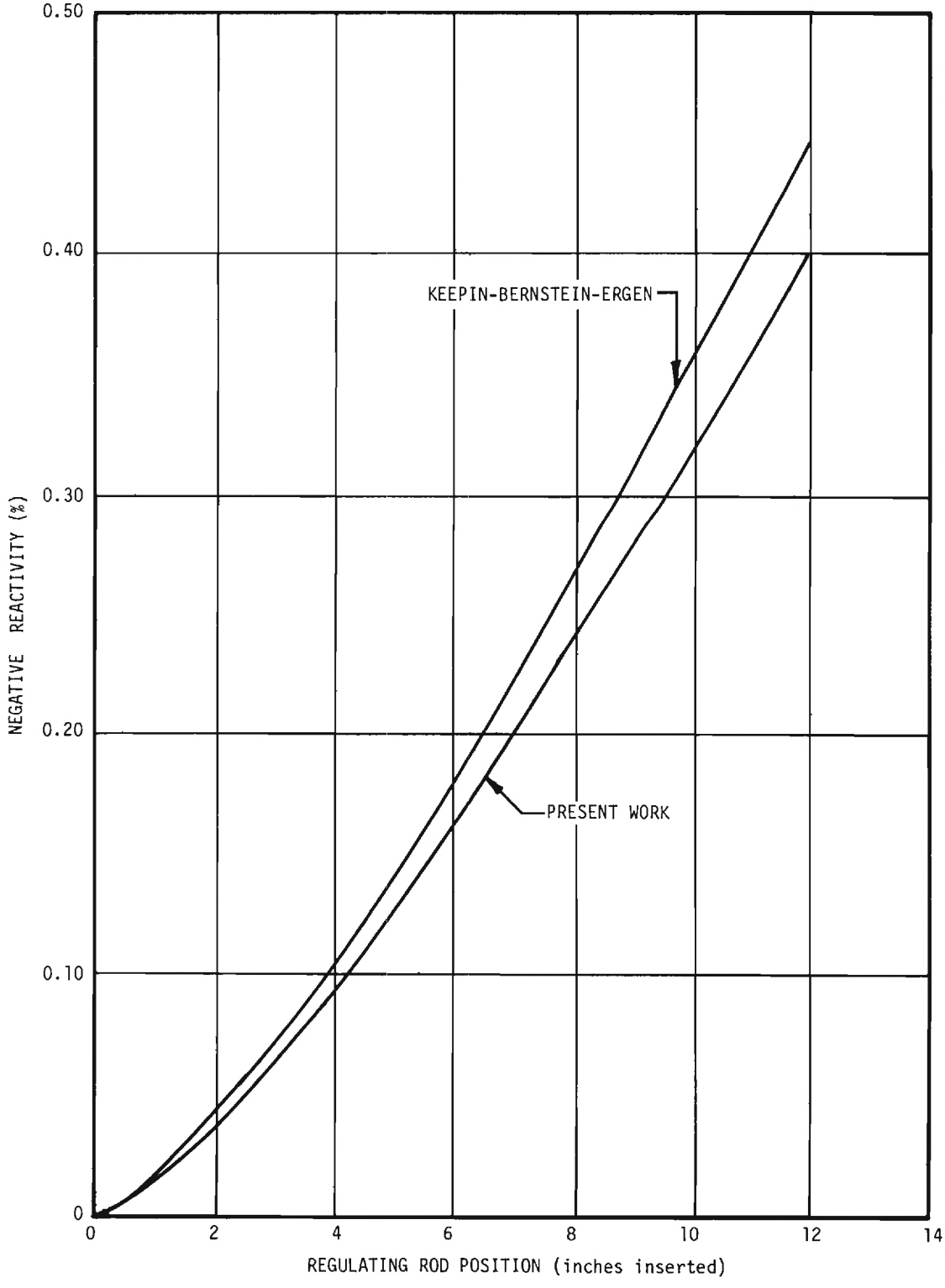


Figure 22. Georgia Tech Research Reactor -- Regulating Rod Calibration as Determined from Experimental Data by Using Both Previously Available Parameters and Those Determined in the Present Work

However, it has been mentioned in the previous chapter that calculations by Lewins[26] at MITR have indicated that βeff for the MITR, which is a highly enriched heavy-water research reactor similar to the GTRR, may be very close to the β value suggested by Keepin.[11]

In the asymptotic period problem, these same two sets of parameters were applied in turn to calculate the steady rising reactor period which would be associated with a wide range of positive reactivity step insertions. Computer program NE-1 was used for these calculations.[37] The results are represented in Figure 23. In this figure we have chosen to plot positive reactivity as a function of the reciprocal of asymptotic period since this is an effective way to emphasize the difference which the choice of parameters makes over a wide range of asymptotic periods. Once again, a reactivity calibration which is obtained using the asymptotic period method and KBE parameters would indicate approximately 10 percent greater worth than would a calibration using the same period data and our parameters. The basic reason for these differences in kinetics calculation results is that, for GTRR applications, the KBE parameters overestimate the photoneutron contribution and simultaneously do not properly account for the enhanced importance of the standard delayed neutrons. Since, on the whole, the photoneutrons with which we are concerned here have longer half-lives than do the other delayed neutrons, KBE parameter calculations may be expected to accentuate the "sluggish" effect which photoneutrons have on reactor transients. With a given set of GTRR data, such as the RODCALIB data previously discussed, the KBE parameter calculation compensates for overestimated sluggishness by overestimating the reactivity.

The detection and data recording equipment employed in this work

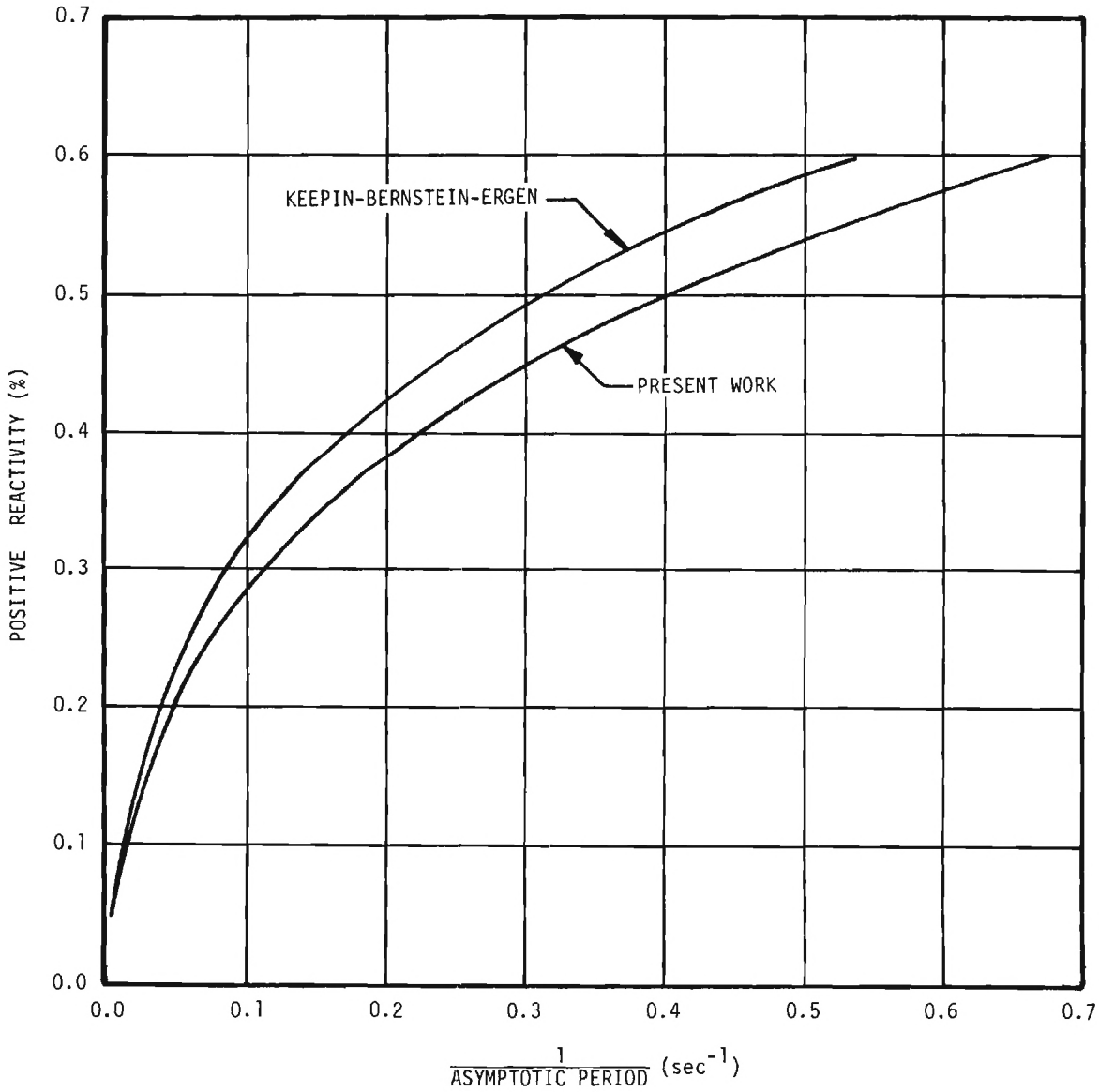


Figure 23. Positive Reactivity as a Function of the Reciprocal of Asymptotic Period Calculated by Using Previously Available and Present Work Parameters

were of the most modern design. Two different neutron detector types, a scintillation detector and a fission chamber, were used. Both detection systems possessed 530 nanosecond pulse resolution times. The experience of this work indicates no consistent advantage of one system over another as far as accuracy and reliability, although precise agreement of results between detectors would have more convincingly validated their equivalence. It is anticipated that, because of their different construction arrangements, the scintillation detector-preamplifier will be more readily affected by radiation damage, and thus have a shorter useful lifetime than the fission chamber assembly.

The purposes of this thesis as set down in Chapter I have been realized. The required number of delayed neutron groups was determined to be 12; relative abundances and decay constants have been provided. A comparison has been made between previously available and presently determined parameters. Furthermore, some of the methods which have been devised in this work offer prospects for useful application to other similar problems in the general area of reactor kinetics.

CHAPTER VI

RECOMMENDATIONS

An immediate extension of this work can be realized by repeating this experiment following higher power operation. The advantage in doing so is that improved statistical precision of data will result. The data collection system described in Chapter II has a resolving time of 530 nanoseconds. The limiting component is the multiscaler. It is anticipated that rather minor improvements of the fission chamber detector will provide for it a resolution of less than 100 nanoseconds. It is, therefore, recommended that this system be improved for higher count rates by the addition of appropriate fast prescalers between the discriminators and the multiscalers. As a matter of operational reliability and reproducibility, it is also recommended that automatic programming of the multiscaler channel dwell time selection be added to the existing equipment.

The values obtained for relative abundances of delayed neutron groups must be related to an appropriate total βeff, $\sum_i \beta_i$eff, in order to be applied to kinetics calculations. A value has been determined in the small-sample experiments previously described which may or may not be appropriate. It is recommended that both calculation and experimental technique be applied to the determination of an absolute value of βeff which is appropriate to a highly enriched heavy-water reactor. The calculational method might also yield values of relative abundance which could be compared with those obtained in the present work. In addition, if an independent experi-

mental value of the prompt neutron lifetime, ℓ^*, were obtained, a direct comparison could be made with the parameter $\beta eff/\ell^*$ as obtained by both noise analysis and pile oscillator methods. The calculation of gamma-ray attenuation by the fuel element in the energy region above the 2.23 MeV threshold energy for photoneutron production in heavy water would provide additional understanding of the details of this process and permit more accurate prediction of the adjustment of abundance parameters necessary for different types of fuel elements.

This work has demonstrated the usefulness of the Variable Metric Minimization method for curve fitting. Methods for its efficient operation should be tested further and optimized techniques for speeding its convergence should be developed.

APPENDICES

APPENDIX A

EXPERIMENTAL RESULTS

The following pages in this appendix include figures which illustrate the success of the curve-fitting operation for Tests A(s) and B(f). Flux decay data were actually accumulated over different time intervals, Δt_j, varying from 0.1 to 1100 seconds. These data values have been divided by the appropriate time, Δt_j, to obtain the count rates represented in Figures 24 and 25. The improved statistical precision associated with a higher total count is evident in these figures; as the collection time per channel increases in Figures 24 and 25, the actual count rates are more consistently described by the computed fit.

In Figures 26 and 27, the value of the weighted residual for each data point is plotted. This residual is the difference between the observed and computed-fit count, divided by the square root of the observed value. From these figures it may be seen that, in both of the tests illustrated, roughly two-thirds of the data points are within one standard deviation of the computed value, and the differences between actual and computed values are randomly positive and negative. If one were to square each of the values indicated in Figure 26, sum up these squares, and then divide by the number of data points (349) minus the number of parameters (26), the result would be the figure-of-merit, V, for this test.

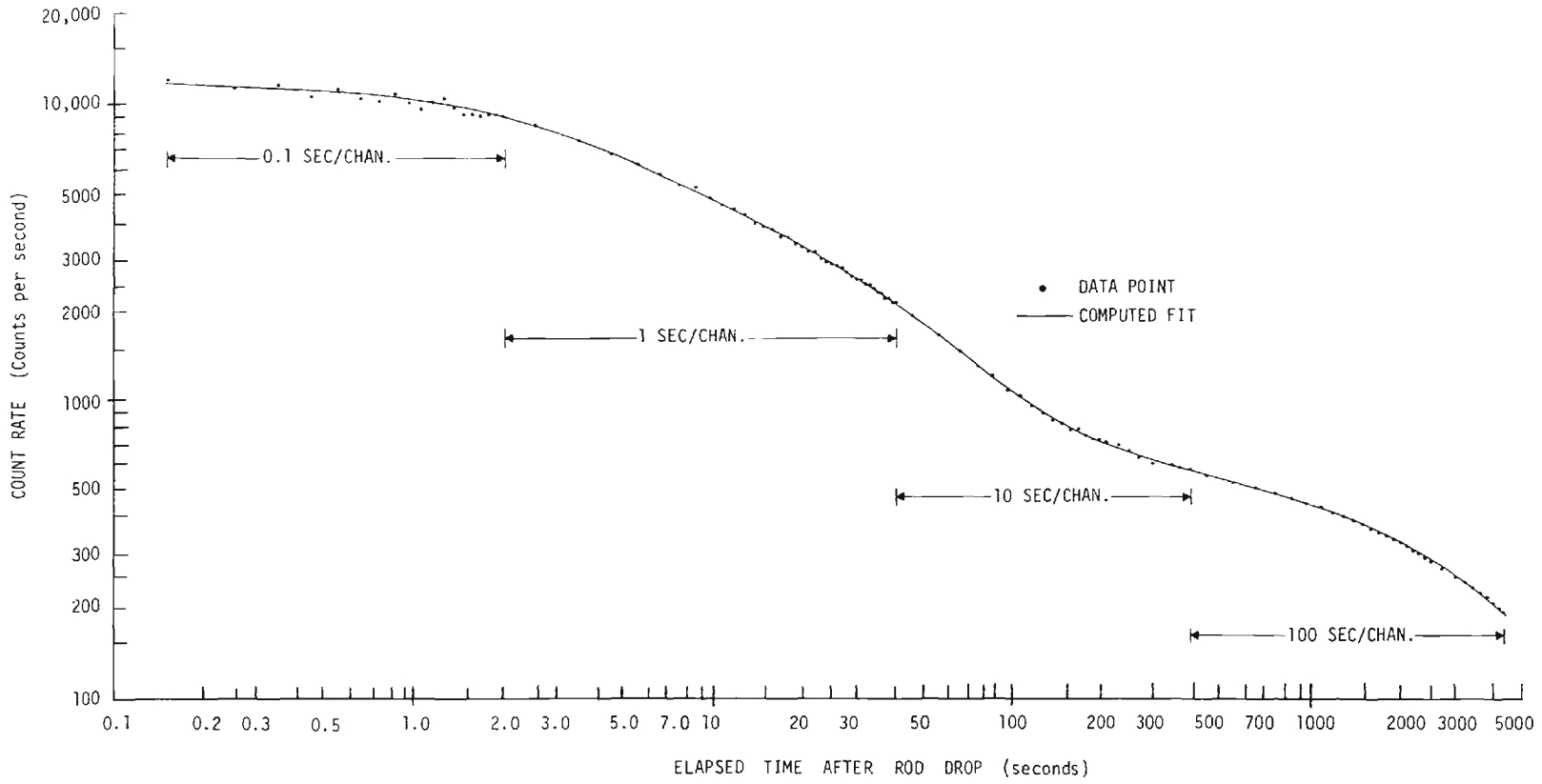


Figure 24a. Representation of Test A(s) Data and Calculated 13-Group Fit from 0.1 - 4000 Seconds (some data points omitted for clarity)

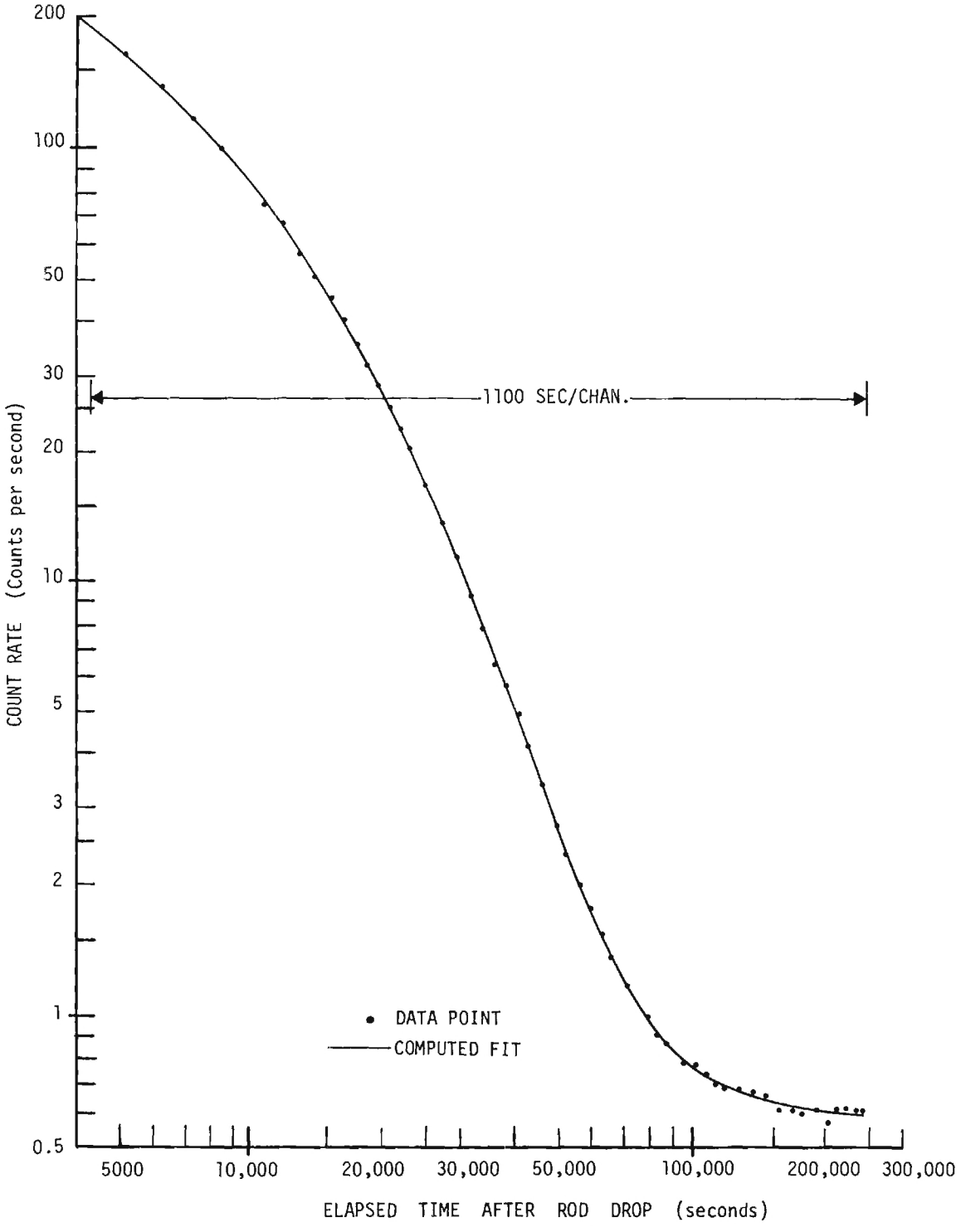


Figure 24b. Representation of Test A(s) Data and Calculated 13-Group Fit from 4000 - 250,000 Seconds (some data points omitted for clarity)

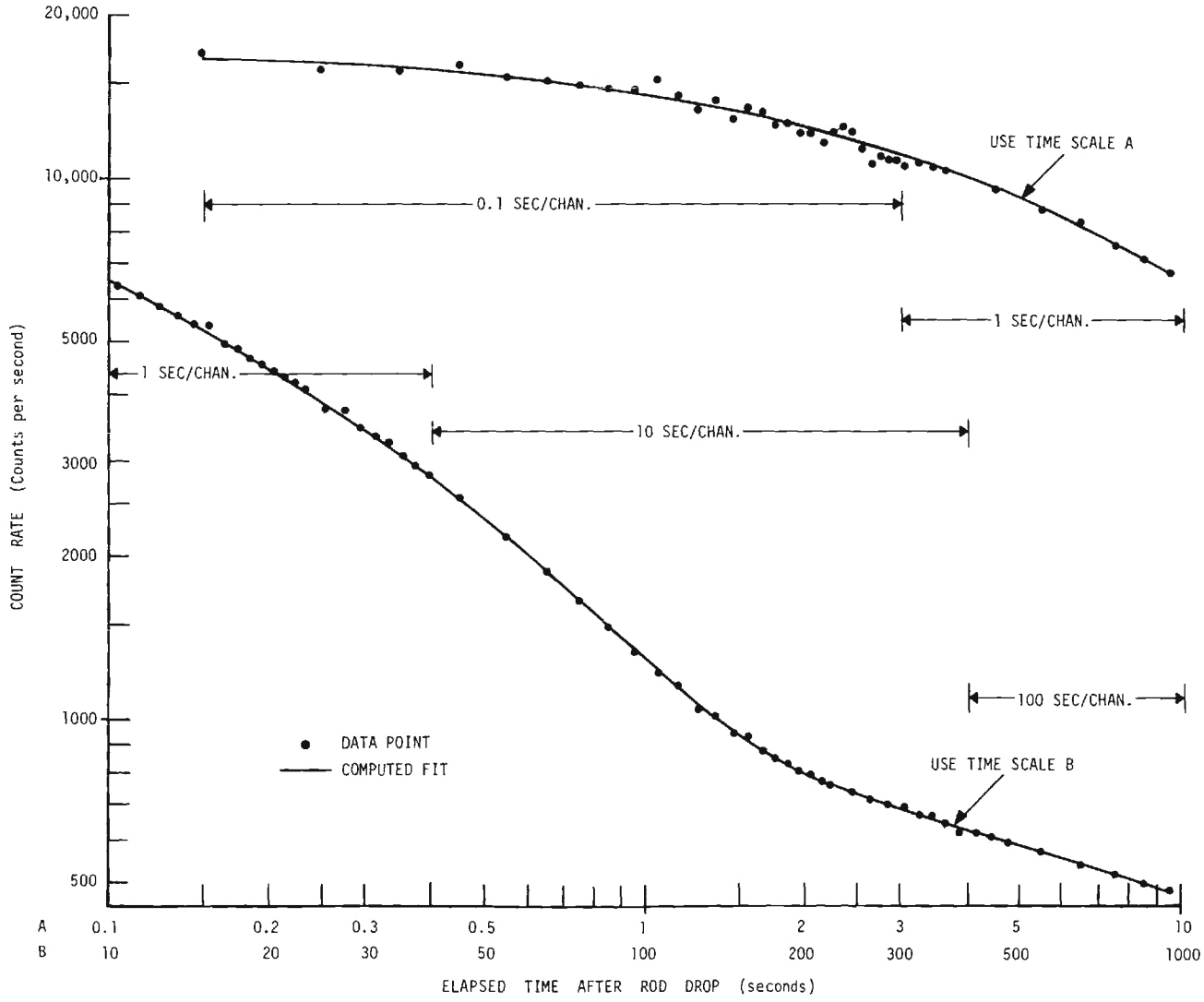


Figure 25a. Representation of Test B(f) Data and Calculated 13-Group Fit from 0.1 - 1000 Seconds (some data points omitted for clarity)

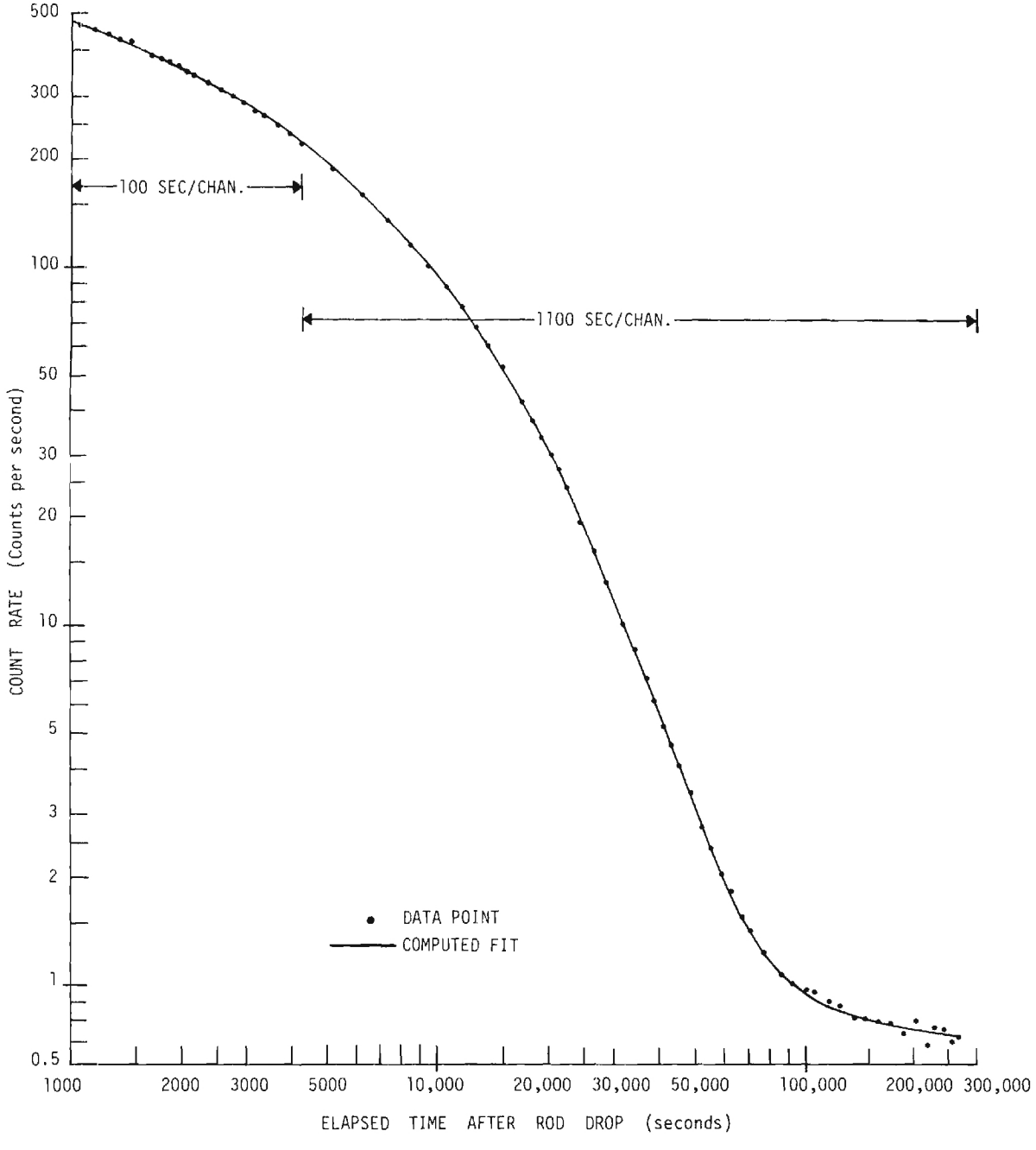


Figure 25b. Representation of Test B(f) Data and Calculated 13-Group Fit from 1000 - 275,000 Seconds (some data points omitted for clarity)

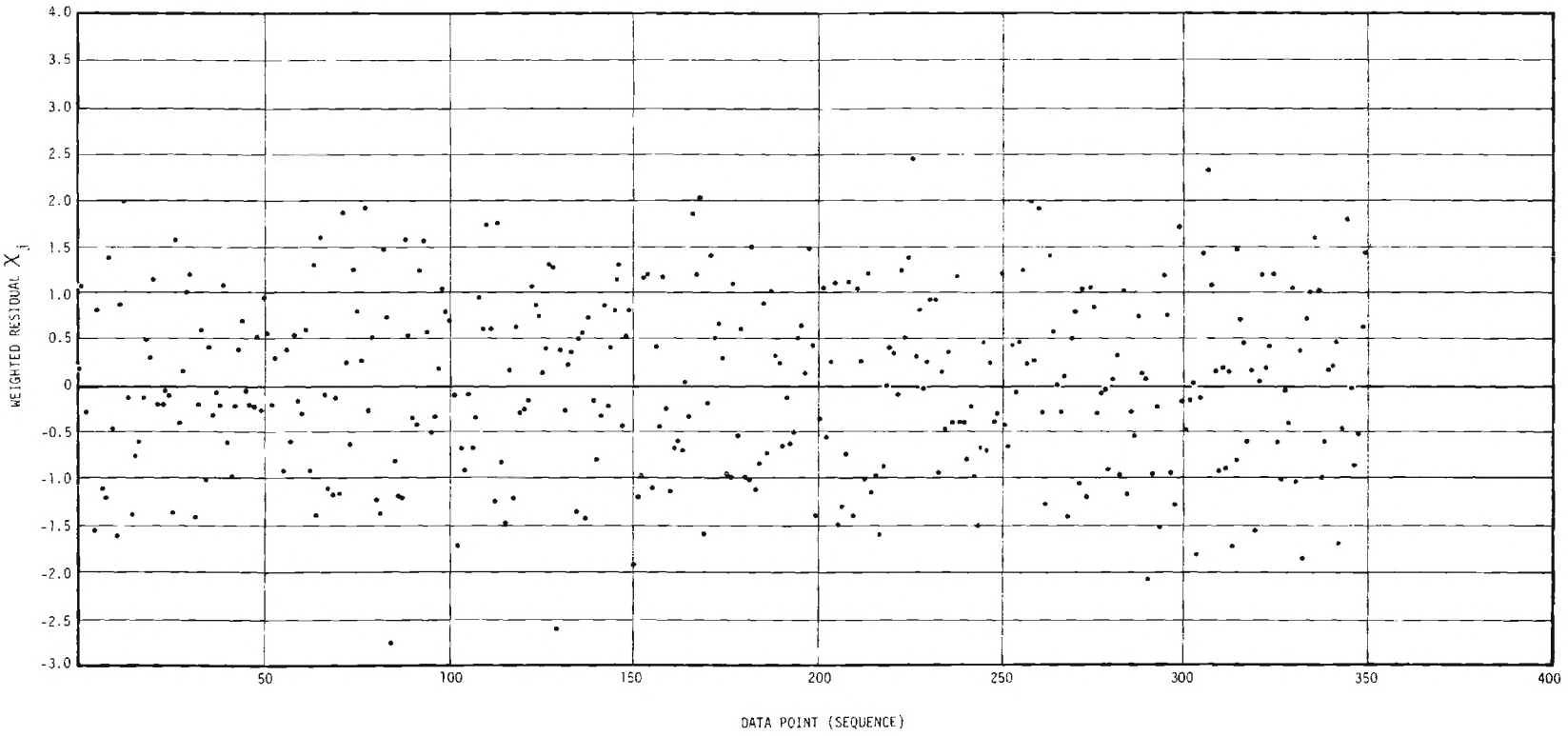


Figure 26. Test A(s) Residuals for Each Data Point in the 13-Group Fit

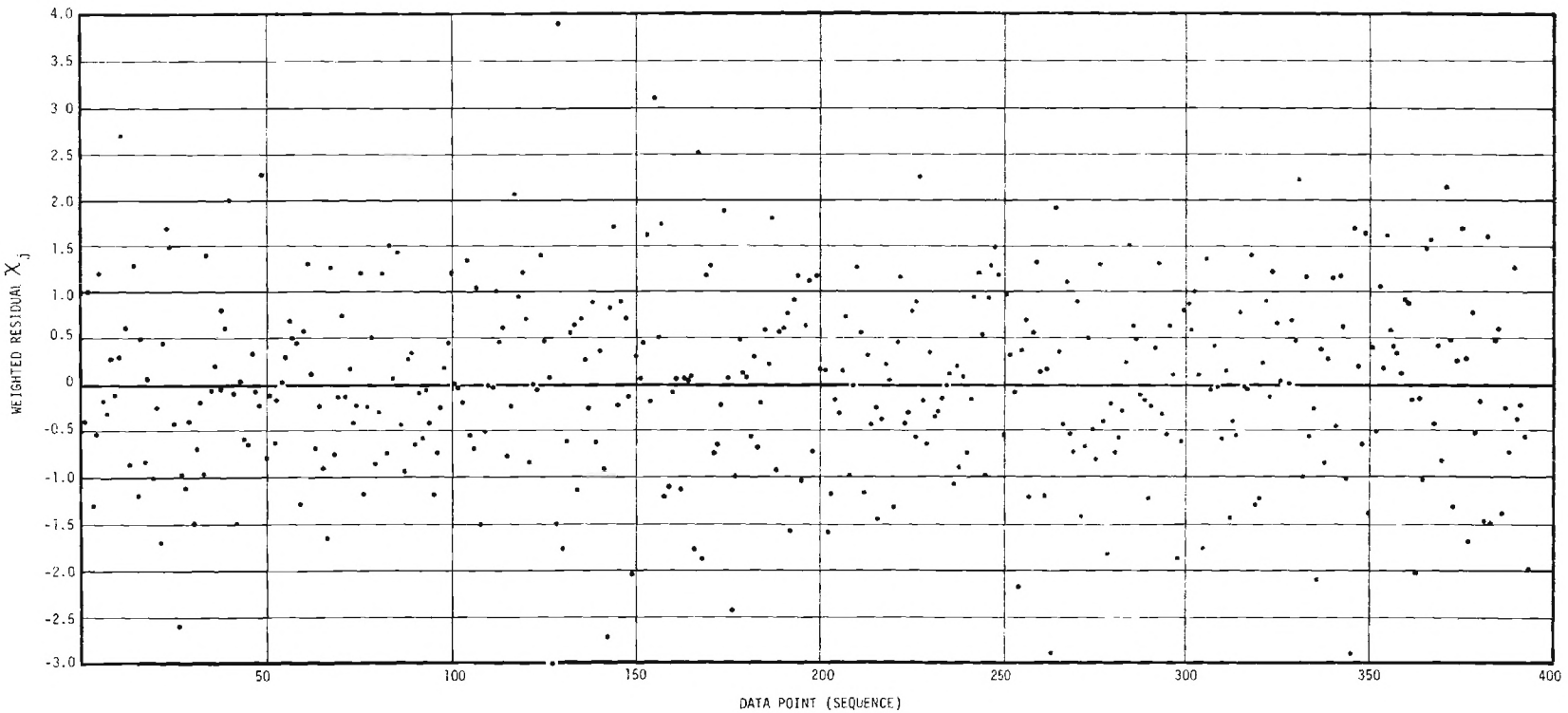


Figure 27. Test B(f) Residuals for Each Data Point in the 13-Group Fit

APPENDIX B

PROCEDURE BLDUP

BLDUP is a B-5500 computer program written in the ALGOL language. It analyzes the power history of the reactor from initial startup to any later time in order to assess the equilibrium fraction of delayed neutron precursors. This equilibrium fraction is expressed as a percent of the saturation value which is associated with the power level existing at the time of interest. In the work reported in this thesis, the time of interest is a specific rod drop time, or alternatively the time at which a control rod calibration by the digital method, RODCALIB,[30] was commenced.

The input data for this program include information taken from the log n recording of reactor power. The recorder receives its signal from a micromicroammeter which operates in conjunction with a compensated ionization chamber located in one of the reactor instrument positions. The ionization chamber is a General Electric Model No. GE-5467870-GR11; the micromicroammeter is General Electric Catalog No. 534E745G3.

Appropriate intervals of reactor operation as recorded on the log n chart are categorized by either exponential or linear approximations and the time and current values are recorded on coding sheets for punched card entries. The only other input quantities required are the decay constants of the delayed neutron precursors, which are determined by the Variable Metric Minimization curve fitting program.

To determine the sequential history of the reactor neutron population, we begin with the standard kinetics equation which describes the

change in delayed neutron precursor concentration with time at a fixed position in the reactor:

$$\frac{dC_i(t)}{dt} = - \lambda_i C_i(t) + \frac{k}{p} \beta_i \Sigma_a \phi(t)$$

By use of an integrating factor, $e^{\lambda_i t}$, this differential equation may be rewritten as

$$\frac{d[C_i(t)\, e^{\lambda_i t}]}{dt} = \frac{k}{p} \beta_i \Sigma_a \phi(t)\, e^{\lambda_i t}$$

and integrated to obtain

$$\int d\left[C_i(t')\, e^{\lambda_i t'}\right] = \int \frac{k}{p} \beta_i \Sigma_a \phi(t')\, e^{\lambda_i t'}\, dt'$$

Now we define $C_i(t=0) = C_{i_o}$, and t_d as the rod drop time of interest, and perform the integration,

$$C_i(t_d) = C_{i_o} e^{-\lambda_i t_d} + \frac{k}{p} \beta_i \Sigma_a e^{-\lambda_i t_d} \int_{t'=o}^{t'=t_d} \phi(t')\, e^{\lambda_i t'}\, dt' \qquad (3)$$

Now, the reference interval is that interval at level power immediately preceding the initiation of an experiment (e.g. rod drop). For the reference case $\phi(t) = \phi^R$, a constant. The equilibrium fraction desired for

each precursor is the number of precursor atoms present under actual conditions divided by the number that would have been present had the reactor been operating at the reference level for an infinite length of time. Therefore we obtain the reference concentration,

$$C_i^R(t_d) = \frac{k}{p}\frac{\beta_i}{\lambda_i}\Sigma_a \phi^R \tag{4}$$

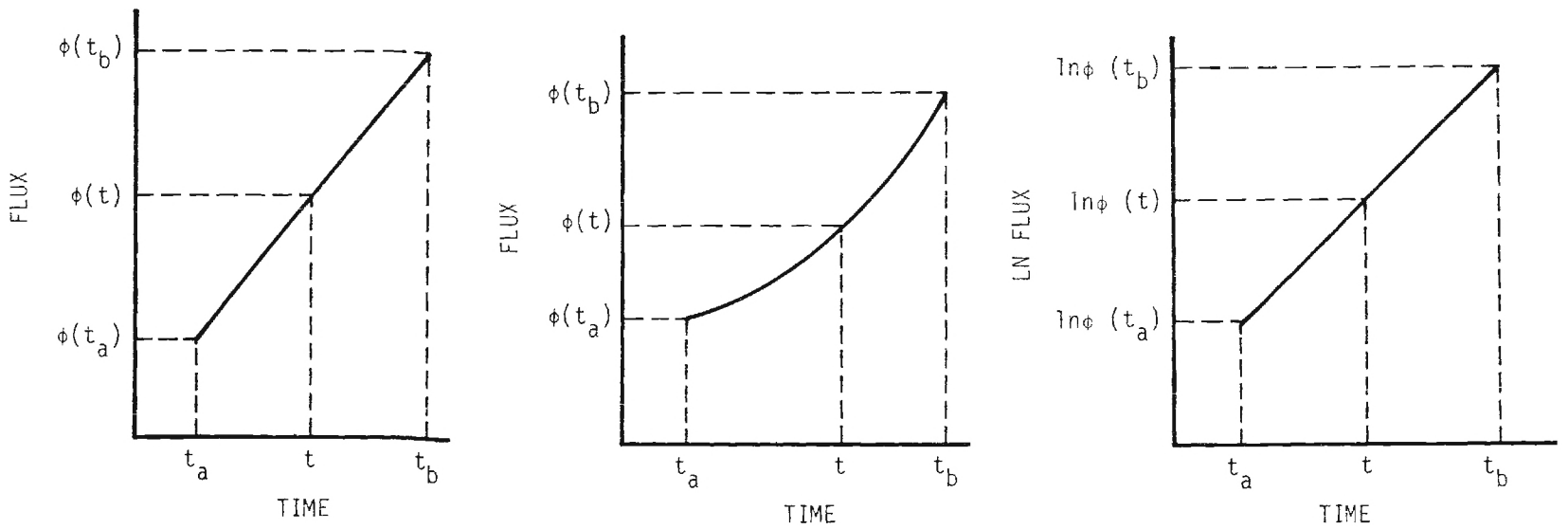
The actual concentration is expressed in Equation 3 above. The equilibrium fraction is:

$$Q_i(t_d) = \frac{C_i(t_d)}{C_i^R(t_d)} \tag{5}$$

The problem then reduces to one of finding the value of the integral in Equation 3 for the exponential or linear portions which make up the integral. In the linear case, referring to Figure 28a, we make the following formulation:

$$\frac{\phi(t) - \phi(t_a)}{t - t_a} = \frac{\phi(t_b) - \phi(t_a)}{t_b - t_a}$$

$$\phi(t) = \phi(t_a) + \frac{t - t_a}{t_b - t_a}\left[\phi(t_b) - \phi(t_a)\right]$$



a. Linear Flux Change Representation

b. Exponential Flux Change Representation

c. Logarithmic Representation of Exponential Flux Change

Figure 28. Standardized Approximations for Reactor Power-History Record

$$\phi(t) = \frac{t_b\,\phi(t_a) - t_a\,\phi(t_b)}{t_b - t_a} + \frac{[\phi(t_b) - \phi(t_a)]\,t}{t_b - t_a}$$

Evaluate $\int_{t'=t_a}^{t'=t_b} \phi(t')\, e^{\lambda_i t'}\, dt'$, with the substitution of $\phi(t)$ developed above for $\phi(t')$ in the integral. Define the value of this integral to be **L**.

$$L = \frac{[t_b\,\phi(t_a) - t_a\,\phi(t_b)]}{t_b - t_a} \int_{t'=t_a}^{t'=t_b} e^{\lambda_i t'}\, dt'$$

$$+ \frac{[\phi(t_b) - \phi(t_a)]}{t_b - t_a} \int_{t'=t_a}^{t'=t_b} t'\, e^{\lambda_i t'}\, dt'$$

$$L = \frac{[t_b\,\phi(t_a) - t_a\,\phi(t_b)]}{t_b - t_a} \frac{1}{\lambda_i} (e^{\lambda_i t_b} - e^{\lambda_i t_a})$$

$$+ \left\{ \frac{[\phi(t_b) - \phi(t_a)]}{(t_b - t_a)} \right\} \left\{ \frac{(t_b\, e^{\lambda_i t_b} - t_a\, e^{\lambda_i t_a})}{\lambda_i} + \frac{(e^{\lambda_i t_a} - e^{\lambda_i t_b})}{\lambda_i^2} \right\}$$

Similarly, in the exponential case illustrated in Figures 28b and 28c, we obtain the following representation:

$$\phi(t) = \phi(t_a)\, e^{\alpha(t - t_a)}$$

where α is some appropriate coefficient to describe the exponential change. Then

$$\ln \frac{\phi(t)}{\phi(t_a)} = \frac{t - t_a}{t_b - t_a} \ln \frac{\phi(t_b)}{\phi(t_a)}$$

$$\phi(t) = \phi(t_a) \left[\frac{\phi(t_b)}{\phi(t_a)} \right]^{\frac{t - t_a}{t_b - t_a}}$$

Defining the value of the integral,

$$\int_{t'=t_a}^{t'=t_b} \phi(t') \, e^{\lambda_i t'} \, dt'$$

as E in this exponential case, we obtain

$$E = \phi(t_a) \int_{t'=t_a}^{t'=t_b} \left[\frac{\phi(t_b)}{\phi(t_a)} \right]^{\frac{t' - t_a}{t_b - t_a}} e^{\lambda_i t'} \, dt'$$

We solve this by substitution as follows:

Let $y = \frac{t'}{t_b - t_a}$. Then $t' = y(t_b - t_a)$ and $dt' = (t_b - t_a)dy$

$$E = \phi(t_a)\left[\frac{\phi(t_b)}{\phi(t_a)}\right]^{\frac{-t_a}{t_b - t_a}}(t_b - t_a)\int_{y=\frac{t_a}{t_b-t_a}}^{y=\frac{t_b}{t_b-t_a}}\left[\frac{\phi(t_b)}{\phi(t_a)}\right]^{y} e^{\lambda_i(t_b-t_a)y}\,dy$$

The integral is of the form, $\int a^y e^{by}\,dy$, or $\int e^{y\ln a} e^{by}\,dy$, which reduces to $\int e^{(\ln a + b)y}\,dy$ and is integrated to produce

$$\left|\frac{e^{(\ln a + b)y}}{(\ln a + b)}\right|_{\frac{t_a}{t_b-t_a}}^{\frac{t_b}{t_b-t_a}}$$

Transforming back to the variables of interest and collecting terms, we obtain:

$$E = (t_b - t_a)\,\phi(t_a)\left[\frac{1}{\ln\frac{\phi(t_b)}{\phi(t_a)} + \lambda_i(t_b - t_a)}\right]\left[\frac{\phi(t_b)}{\phi(t_a)} e^{\lambda_i t_b} - e^{\lambda_i t_a}\right]$$

We have thus demonstrated the method for evaluating the integral in Equation 3 if the power history record is broken up into intervals approximating either linear or exponential changes in flux.

In evaluating Equations 3 and 4 it is not necessary to assess the value of the terms k, p, β_i, and Σ_a since they are always present in

both numerator and denominator. For the calculation from startup, the C_{i_o} values are all zero. For subsequent analyses, starting at the rod drop time of the previous analysis and calculating forward in time, the values of C_{i_o} are simply the values of $C_i(t_d)$ which were calculated for the last rod drop time if the decay constants remain unchanged. It is evident that successive analyses must be contiguous in time.

The computer procedure, BLDUP, delivers in printed and punched form the values of equilibrium fractions according to Equation 5 and also punches the values of $C_i(t_d)$ for use in the next succeeding analysis.

The listing of this procedure is reproduced, beginning on the following page.

```
COMMENT   B L O U P ;                                                    0000
 BEGIN FILE IN CARD 0 (2, 15) ; %                                  100   0000
                                                  START OF SEGMENT ********** 0002
 FILE OUT PUNCH 0 (2, 10) ; %                                      200   0005
 FILE OUT PRINT 1 (2, 15) ; %                                      300   0010
 INTEGER NP, I, CONIND, TPREV, TCON, NOINT, K, TD, Z, M, TSTART;   400   0015
 INTEGER ARRAY TYPE [0 : 1022], T [0 : 1022];                      500   0015
 REAL PHID, SUMINT, FAC1, FAC2, FAC3, FAC5, FAC6, FAC7, L, E;      600   0019
 REAL ARRAY LAM [0 : 20], CZERO [0 : 20], PHI [0 : 1022], CR [0 :  700   0019
 20], THA [0 : 20], TTMIN [0 : 20], CT [0 : 20], EQFRAC [0 : 20];  800   0025
 LABEL L1, LB, L2, L3, L4, LA, L5, L6, L7;                         900   0035
 ALPHA C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12, C13, C14; 1000 0035
 FORMAT FL1 (                                                      1100  0035
                                                  START OF SEGMENT ********** 0003
 "        BLDUP, A B-5500 ALGOL PROCEDURE FOR CALCULATING THE              0035
CONCENTRATION OF DELAYED NEUTRON PRECURSORS                      "         0035
 ), FL2 (14 A5), FL3 (/ / / / "DATE OF THIS CALCULATION IS ", 14 A5  1300  0035
 ), FL4 (I10 / (6 E12. 5)), FL5 ("  I              LAMBDA         "),  1400  0035
 FL6 (/ X2, I2, X12, R10. 5), FL7 (3 I10), FL8 (/ /                  1500  0035
 " THE PREVIOUS BUILDUP ANALYSIS ENDED AT ", I6, /                   1600  0035
 " THIS ANALYSIS BEGINS AT ", I6), FL9 (6 E12. 5), FL10 (I10), FL11  1700  0035
 (I12, I12, R12. 4), FL12 (I10, R12. 4), FL13 (I10, X10, R12. 4);    1800  0035
                                                  0003 IS 0118 LONG, NEXT SEG 0002
 LIST LIST1 (C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12, C13  1900  0035
 , C14) ; %                                                          2000  0052
 LIST LIST2 (NP, FOR I ← 1 STEP 1 UNTIL NP DO LAM [I]);              2100  0058
 LIST LIST3 (I, LAM [I]) ; %                                         2200  0069
 LIST LIST4 (CONIND, TPREV, TCON) ; %                                2300  0077
 LIST LIST5 (TPREV, TCON) ; %                                        2400  0086
 LIST LIST6 (FOR I ← 1 STEP 1 UNTIL NP DO CZERO [I]);                2500  0094
 LIST LIST10 (NOINT) ; %                                             2600  0104
 LIST LIST11 (FOR K ← 1 STEP 1 UNTIL NOINT DO [TYPE [K], T [K], PHI  2700  0110
 [K]]) ; %                                                           2800  0116
 LIST LIST12 (TD, PHID) ; %                                          2900  0123
 LIST LIST13 (FOR I ← 1 STEP 1 UNTIL NP DO [I, EQFRAC [I]]);         3000  0131
 LIST LIST14 (FOR I ← 1 STEP 1 UNTIL NP DO CT [I]);                  3100  0143
 WRITE (PRINT [PAGE]) ; %                                            3200  0153
 WRITE (PRINT, FL1) ; %                                              3300  0156
 READ (CARD, FL2, LIST1) ; %                                         3400  0159
 WRITE (PRINT, FL3, LIST1) ; %                                       3500  0163
 READ (CARD, FL4, LIST2) ; %                                         3600  0167
 WRITE (PRINT, FL5) ; %                                              3700  0171
 FOR I ← 1 STEP 1 UNTIL NP DO WRITE (PRINT, FL6, LIST3);             3800  0175
 READ (CARD, FL7, LIST4) ; %                                         3900  0184
 WRITE (PRINT, FL8, LIST5) ; %                                       4000  0188
```

```
READ (CARD, FL9, LIST6) ; %                                    4100   0191
READ (CARD, FL10, LIST10) ; %                                  4200   0195
READ (CARD, FL11, LIST11) ; %                                  4300   0199
READ (CARD, FL12, LIST12) ; %                                  4400   0203
FOR K ← 1 STEP 1 UNTIL NOINT DO T [K] ← T [K] × 60;            4500   0207
TO ← TO × 60 ; %                                               4600   0218
I ← 0 ; %                                                      4700   0219
L1: I ← I + 1 ; %                                              4800   0220
IF (I > NP) THEN GO TO L8 ; %                                  4900   0222
T [NOINT + 1] ← TO ; %                                         5000   0223
PHI [NOINT + 1] ← PHIO ; %                                     5100   0225
CR [I] ← (PHIO / LAM [I]) ; %                                  5200   0227
THA [I] ← 0.693 / LAM [I] ; %                                  5300   0229
TTMIN [I] ← TO - 8 × THA [I] ; %                               5400   0231
SUMINT ← 0 ; %                                                 5500   0233
Z ← NOINT ; %                                                  5600   0234
IF (TTMIN [I] ≤ TCON) THEN GO TO L4;                           5700   0235
L2: IF (TTMIN [I] ≥ T [Z]) THEN GO TO L3;                      5800   0236
Z ← Z - 1 ; %                                                  5900   0238
GO TO L2 ; %                                                   6000   0240
L3: TSTART ← T [Z] ; %                                         6100   0242
GO TO LA ; %                                                   6200   0243
L4: TSTART ← TCON ; %                                          6300   0243
Z ← 1 ; %                                                      6400   0244
LA: FOR M ← Z STEP 1 UNTIL NOINT DO                            6500   0245
BEGIN IF (TYPE [M] ≠ 0) THEN GO TO L6;                         6600   0247
  L5: FAC1 ← T [M + 1] - T [M] ; %                             6700   0248
  FAC2 ← EXP (LAM [I] × (T [M + 1] - TO));                     6800   0251
  FAC3 ← EXP (LAM [I] × (T [M] - TO));                         6900   0254
  FAC5 ← (T [M + 1] × PHI [M] - T [M] × PHI [M + 1]) / FAC1;   7000   0257
  FAC6 ← FAC2 - FAC3 ; %                                       7100   0262
  FAC7 ← (PHI [M + 1] - PHI [M]) / FAC1;                       7200   0263
  L ← FAC5 × FAC6 / LAM [I] + FAC7 × ((T [M + 1] × FAC2 - T [M] ×  7300   0266
  FAC3) / LAM [I] - FAC6 / LAM [I] × LAM [I]);                 7400   0269
  SUMINT ← SUMINT + L ; %                                      7500   0274
  GO TO L7 ; %                                                 7600   0275
  L6: FAC1 ← T [M + 1] - T [M] ; %                             7700   0276
  FAC2 ← EXP (LAM [I] × (T [M + 1] - TO));                     7800   0278
  FAC3 ← EXP (LAM [I] × (T [M] - TO));                         7900   0281
  FAC6 ← LN (PHI [M + 1] / PHI [M]);                           8000   0284
  E ← FAC1 × PHI [M] × (1 / (FAC6 × LAM [I] × FAC1)) × ((PHI [M +  8100   0287
  1] / PHI [M]) × FAC2 - FAC3) ; %                             8200   0290
  SUMINT ← SUMINT + E ; %                                      8300   0293
```

```
  L7:                                                                    8400    0295
END ; %                                                                  8500    0295
CT [I] ← CZERO [I] × EXP (- LAM [I] × (TSTART - TPREV)) + SUMINT;        8600    0297
EQFRAC [I] ← CT [I] / CR [I] ; %                                         8700    0301
GO TO L1 ; %                                                             8800    0304
L8: WRITE (PRINT, FL13, LIST13) ; %                                      8900    0304
WRITE (PUNCH, FL9, LIST14) ; %                                           9000    0308
END.                                                                     9100    0312
```

APPENDIX C

RP-129 AND RODROP

RP-129 is a FORTRAN program which solves the one-group, space-independent reactor kinetics equations for neutron density if the reactor kinetics parameters and the excess reactivity as a function of time are given. A complete description of the method of solution is given by Kaganove.[31] This program has been translated into ALGOL for the B-5500. It permits several optional functional representations of reactivity as a function of time plus an interrupt feature which enables a redefining of reactivity function in the middle of the transient solution for neutron flux.

RP-129 has been modified somewhat in order to provide punched output which can be efficiently used in a second program, RODROP,[32] which has also been translated into ALGOL. RP-129 provides, at selected times, the relative neutron density and current value of delayed neutron precursor concentration. These quantities, along with the kinetics parameters, represent the input to RODROP. It is the purpose of RODROP to ascertain at what time during a non-instantaneous insertion of reactivity the instantaneous insertion of the same amount of reactivity would be most nearly equivalent, in terms of resulting neutron density.

Following the insertion of a fixed amount of reactivity, Δk remains constant, a static reactor configuration exists, and the reactor flux may

be expressed as

$$\Phi(t) = \sum_{i=1}^{N} a_i e^{s_i(t-T)}$$

where T = reactivity insertion time

t = elapsed time.

The s_i are inhour equation roots and are found in the same fashion as in References 8 and 37. The a_i may be calculated from the values of flux and precursor concentration provided by RP-129.

If the time for reactivity insertion is not instantaneous, the value of T to be used for calculational purposes is questionable, since neither the start of drop nor the stop of the drop are necessarily the "best time."

The conclusion that, for some long time after a reactivity insertion, instantaneous or otherwise, the detailed history of the insertion should have little effect on the flux leads to the proposal that a time T′ be chosen such that

$$a_1 e^{s_1(t-T)} = a_1' e^{s_1(t-T')}$$

Here, s_1 is the smallest in magnitude of the s_i (and is associated with the longest half-life of interest) and the a_1' is the coefficient which is found in the analysis of an instantaneous drop. The output of this program is the constructive time of insertion of a Δk in a stepwise manner which is most nearly equivalent at a later time to the true insertion. In investi-

gating results from these two programs, it has been found by Cohn[32] that, for a linear reactivity insertion, the ratio of T/T' is always less than 0.5 and approaches that maximum as the absolute value of the Δk decreases. The equivalent instantaneous drop flux predicted by the RODROP program is always less than that of the actual non-instantaneous insertion.

These programs in the ALGOL language are included for reference in the following pages.

```
COMMENT   R P - 1 2 9 ;                                                    0000
 BEGIN FILE OUT PUNCH 0 (2, 10) ; %                                  100    0000
                                                  START OF SEGMENT ********** 0002
 FILE TAPE6 1 (2, 15) ; %                                            200    0005
 FILE TAPE7 0 (2, 15) ; %                                            300    0010
 BOOLEAN ARRAY SENSL [0 : 4] ; %                                     400    0015
 REAL Q, XPR ; %                                                     500    0017
 INTEGER K ; %                                                       600    0017
 REAL PROCEDURE INT (ARG1) ; %                                       700    0017
 VALUE ARG1 ; %                                                      800    0017
 REAL ARG1 ; %                                                       900    0017
 BEGIN INT ← (SIGN (ARG1) × ENTIER (ABS (ARG1)))                     1000   0017
 END ; %                                                             1100   0020
 REAL PROCEDURE MAX (ARG1, ARG2) ; %                                 1200   0025
 VALUE ARG1, ARG2 ; %                                                1300   0025
 REAL ARG1, ARG2 ; %                                                 1400   0025
 BEGIN MAX ← (IF ARG1 ≥ ARG2 THEN ARG1 ELSE ARG2)                    1500   0025
 END ; %                                                             1600   0027
 REAL PROCEDURE MIN (ARG1, ARG2) ; %                                 1700   0030
 VALUE ARG1, ARG2 ; %                                                1800   0030
 REAL ARG1, ARG2 ; %                                                 1900   0030
 BEGIN MIN ← (IF ARG1 ≤ ARG2 THEN ARG1 ELSE ARG2)                    2000   0030
 END ; %                                                             2100   0032
 PROCEDURE SRASYMPR (JREX, JALPHA, JMAXI, SVBETA, SVFLAM, JBETASU,   2200   0035
 JVITA, JSOURCE) ; %                                                 2300   0035
 INTEGER JMAXI ; %                                                   2400   0035
 REAL JREX, JALPHA, JBETASU, JVITA, JSOURCE;                         2500   0035
 REAL ARRAY SVBETA [0], SVFLAM [0];                                  2600   0035
 BEGIN INTEGER JI ; %                                                2700   0035
                                                  START OF SEGMENT ********** 0003
   REAL JR, JSUM, JALPHAG, JSUM2, JX, JY;                            2800   0000
   LABEL DUMMY, L1, L10, L20 ; %                                     2900   0000
   IF JREX > 0 THEN GO TO L1 ; %                                     3000   0000
   JALPHA ← 0 ; %                                                    3100   0001
   GO TO DUMMY ; %                                                   3200   0002
   COMMENT  CALCULATE INITIAL GUESS BY EMPIRICAL FORMULA;            3300   0002
   L1: JR ← 1 + JREX ; %                                             3400   0002
   JSUM ← 0 ; %                                                      3500   0004
   JI ← 1 ; %                                                        3600   0005
   DO                                                                3700   0005
   BEGIN JSUM ← JSUM + SVBETA [JI] / SVFLAM [JI]                     3800   0005
   END UNTIL (JI ← (JI + 1)) > JMAXI;                                3900   0006
   JALPHAG ← JREX / (JVITA + JR × JSUM / (1 + (.2 × JREX / JBETASU)   4000   0010
   ↑ 2 / SQRT (JVITA))) ; %                                          4100   0012
```

```
  COMMENT  ITERATE USING NEWTONS METHOD UNTIL 6 SIGNIFICANT FIGURES;          4200   0016
  L10: JSUM ← 0 ; %                                                          4300   0016
  JSUM2 ← 0 ; %                                                              4400   0016
  JI ← 1 ; %                                                                 4500   0017
  DO                                                                         4600   0018
  BEGIN JX ← 1 / (JALPHAG + SVFLAM [JI]);                                    4700   0018
    JY ← SVBETA [JI] × SVFLAM [JI] × JX;                                     4800   0020
    JSUM ← JSUM + JY ; %                                                     4900   0022
    JSUM2 ← JSUM2 + JY × JX                                                  5000   0023
  END UNTIL (JI ← (JI + 1)) > JMAXI;                                         5100   0024
  JALPHA ← JALPHAG - (JVITA × JALPHAG - JREX - JK × (JSUM -                   5200   0027
  JBETASU)) / (JVITA + JK × JSUM2);                                          5300   0030
  IF (ABS ((JALPHA - JALPHAG) / JALPHA) ≤ 10-6) THEN GO TO DUMMY;            5400   0033
  JALPHAG ← JALPHA ; %                                                       5500   0035
  GO TO L10 ; %                                                              5600   0036
  L20: GO TO DUMMY ; %                                                       5700   0039
  DUMMY:                                                                     5800   0039
END ; %                                                                      5900   0040

                                                  0003 IS 0044 LONG, NEXT SEG 0002

PROCEDURE MAINPRO ; %                                                        6000   0035
BEGIN REAL ARRAY ARRAY1 [0 : 10] ; %                                         6100   0035

                                                  START OF SEGMENT ********** 0004

  DEFINE SVTEST0 = ARRAY1 # ; %                                              6200   0002
  DEFINE JT = ARRAY1 [1] # ; %                                               6300   0002
  DEFINE JALPHA = ARRAY1 [5] # ; %                                           6400   0002
  DEFINE JTHERM = ARRAY1 [4] # ; %                                           6500   0002
  DEFINE JPHI = ARRAY1 [3] # ; %                                             6600   0002
  DEFINE JREX = ARRAY1 [2] # ; %                                             6700   0002
  REAL ARRAY SVBETA [0 : 20], SVFLAM [0 : 20], SVCIDOT [0 : 20],             6800   0002
  SVADENT [0 : 10], SVP [0 : 100], SVTLIM [0 : 20], SVTLIP [0 : 20           6900   0008
  ], SVRLTI [0 : 20], SVBLTI [0 : 20], SVC [0 : 20], SVPDT [0 :              7000   0014
  100] ; %                                                                   7100   0022
  INTEGER DX1 ; %                                                            7200   0024
  OWN INTEGER ASVKRTN ; %                                                    7300   0024
  INTEGER JIFBLAM, JNUMOPT, JICHECK, JINTRUP, JIFLAG, JIMAX, JI,             7400   0024
  JNUMEXC, JJC, JINTM, JJPRINT, JNUMREX, JNREAD, JK, JJ, JL;                 7500   0024
  REAL JWUERY, JBETASU, JTLAST, JVITA, JS, JEPSILO, JSUM, JRHO,              7600   0024
  JSOURCE, JPERIOD, JENDOT, JENDOT2, JELTATC, JSTEPK, JPHIZ, JW,             7700   0024
  JBLT, JSUMDOT, JSL, JTTEST, JT2, JRT2, JRTSQR, JDELNM1, JDELNM2,           7800   0024
  JT2L, JBLT2, JBLTVIT, JDELTAT, JCBLT, JCBLT2, JCBLT3, JDELPHI,             7900   0024
  JT1, JA, JB, JTRY, JDELK, JDELK3N, JX, JDELK1, JDK1, JDK2, JDK3;           8000   0024
  COMMENT   THE FOLLOWING SUBROUTINES ARE REQUIRED:                          8100   0024
          SRASYMPR ; %                                                       8200   0024
  FORMAT FL1001 (2 I4, 10 A6, X9, I3), FL1 (/                                8300   0024
```

```
                                                                    START OF SEGMENT ********** 0005
"JICHECK TEST FAILED"), FL1005 (112 / (6 R12. 1)), FL1015 (6          8400    0024
R12. 1), FL1060 (6 R12. 1), FL1080 (                                  8500    0024
" HP129 ONE GROUP,SPACE INDEPENDENT REACTOR KINETICS"), FL1085 (      8600    0024
/ " ", 10 A6), FL3003 (" " / 10 A6), FL1090 (/                        8700    0024
" NUMBER OF PRECURSORS=", I2 / X15, "I", X11, "BETA", X10,            8800    0024
"LAMBDA", X7, "L*(C DOT)" / (I16, 3 R16. 6)), FL1095 (/ " ", X5,      8900    0024
"KEX(0-)", X7, "PERIOD(0-)", X6, "NDOT/N (0-)" / 3 R16. 6),           9000    0024
FL1100 (" ", X5, "WHERE NDOT/N (0-) WAS INPUT INFORMATION"),          9100    0024
FL1105 (" ", X5, "WHERE KEX(0-) WAS INPUT INFORMATION"), FL1110 (     9200    0024
" ", X5, "WHERE KEX(0-) AND THE CIDOTS WERE INPUT INFORMATION"),      9300    0024
FL1115 (/ " ", X3, "INITIAL TIME", X4, "FINAL TIME", X8,              9400    0024
"SOURCE", X9, "LIFETIME" / 4 R16. 6), FL1200 (3 I12, 3 R12. 1),       9500    0024
FL1350 (/ " REACTIVITY OPTION", X7, I2 / " REACTIVITY STEP", X5,      9600    0024
R16. 6 / " INTERRUPT OPTION", X9, I2 / " INTERRUPT VALUE", R20.       9700    0024
6 / / " ", (6 R20. 6)), FL1355 (/ " DELTA T=", R12. 4,                9800    0024
",RESULTS PRINTED EVERY", I6, " STEPS."), FL1360 (/ " ", X8,          9900    0024
"TIME", X12, "FLUX", X10, "PERIOD", X9, "INTEGRAL N", X6,            10000   0024
"K EXCESS"), FL1480 (5 R16. 6), FL3001 (6 R20. 10), FL3002 (4         10100   0024
R20. 10) ; %                                                          10200   0024
                                                                    0005 IS 0233 LONG, NEXT SEG 0004
LIST LIST1 (JIFBLAM, JNUMOPT, FOR DX1 ← 1 STEP 1 UNTIL 10 DO          10300   0024
SVADENT (DX1), JICHECK) ; %                                           10400   0029
LIST LIST2 (JIMAX, FOR DX1 ← 1 STEP 1 UNTIL JIMAX DO SVBETA (DX1      10500   0038
), FOR DX1 ← 1 STEP 1 UNTIL JIMAX DO SVFLAM (DX1))%                   10600   0042
LIST LIST3 (JT, JTLAST, JVITA, JALPHA, JS, JPHI, JREX, JEPSILO)%      10700   0054
LIST LIST4 (FOR DX1 ← 1 STEP 1 UNTIL JIMAX DO SVCIDOT (DX1))%         10800   0070
LIST LIST5 (FOR DX1 ← 1 STEP 1 UNTIL 8 DO SVADENT (DX1))%             10900   0080
LIST LIST6 (JIMAX, FOR DX1 ← 1 STEP 1 UNTIL JIMAX DO (DX1,            11000   0090
SVBETA (DX1), SVFLAM (DX1), SVCIDOT (DX1)))%                          11100   0095
LIST LIST7 (JREX, JPERIOD, JALPHA)%                                   11200   0106
LIST LIST8 (JT, JTLAST, JS, JVITA)%                                   11300   0115
LIST LIST9 (JNUMEXC, JINTRUP, JJC, JELTATC, JQUERY, JSTEPK)%          11400   0125
LIST LIST10 (FOR DX1 ← 1 STEP 1 UNTIL JNREAD DO SVP (DX1))%           11500   0138
LIST LIST11 (JNUMREX, JSTEPK, JINTRUP, JQUERY, FOR DX1 ← 1 STEP       11600   0148
1 UNTIL JNREAD DO SVP (DX1)) ; %                                      11700   0154
LIST LIST12 (JDELTAT, JJPRINT) ; %                                    11800   0163
LIST LIST13 (JT) ; %                                                  11900   0171
LIST LIST14 (JT, JPHI, JPERIOD, JTHERM, JREX)%                        12000   0178
LIST LIST15 (FOR DX1 ← 1 STEP 1 UNTIL JIMAX DO SVC (DX1))%            12100   0190
LIST LIST16 (JT, JREX, JPHI, FOR DX1 ← 1 STEP 1 UNTIL JIMAX DO        12200   0200
SVC (DX1)) ; %                                                        12300   0207
LABEL L1, L2, L10, L15, L20, L30, L50, L60, L70, L100, L105,          12400   0214
L110, L115, L200, L210, L220, L235, L245, L5000, L5001, L5002,        12500   0214
```

```
L5003, L5050, L250, L300, L301, L302, L303, L304, L350, L400,          12600   0214
L406, L407, L420, L440, L446, L459, L460, L465, L480, L490,            12700   0214
L2100, L2101, L2102, L2103, L2200, L2210, L2215, L2104, L2105,         12800   0214
L2000, L2001, LEND, L71 ; %                                            12900   0214
SWITCH ASSIGN = L406, L407 ; %                                         13000   0214
SWITCH SWGD1 = L20, L30, L60 ; %                                       13100   0219
SWITCH SWGD2 = L100, L105, L110;                                       13200   0224
SWITCH SWGD3 = L5000, L5001, L5002, L5003;                             13300   0229
SWITCH SWGD4 = L301, L302, L303, L304;                                 13400   0235
SWITCH SWGD5 = L2100, L2102, L2200, L2104;                             13500   0241
SWITCH SWGD6 = L2101, L2103, L2200, L2105;                             13600   0247
L1: READ (TAPE7, FL1001, LIST1) [LEND];                                13700   0253
IF (JICHECK = 129) THEN GO TO L2;                                      13800   0259
GO TO L2001 ; %                                                        13900   0260
L21 JINTRUP = 0 ; %                                                    14000   0262
JIFLAG = 0 ; %                                                         14100   0262
JQUERY = 0 ; %                                                         14200   0263
IF (XPR = (JIFBLAM)) > 0 THEN GO TO L15;                               14300   0264
IF XPR < 0 THEN GO TO L10 ; %                                          14400   0266
READ (TAPE7, FL1005, LIST2) ; %                                        14500   0267
JBETASU = SVBETA [1] ; %                                               14600   0271
JI = 2 ; %                                                             14700   0272
DO                                                                     14800   0273
BEGIN JBETASU = JBETASU + SVBETA [JI]                                  14900   0273
END UNTIL (JI = (JI + 1)) > JIMAX;                                     15000   0273
GO TO L15 ; %                                                          15100   0276
L10: SVBETA [1] = .00025 ; %                                           15200   0279
SVBETA [2] = .00166 ; %                                                15300   0280
SVBETA [3] = .00213 ; %                                                15400   0281
SVBETA [4] = .00241 ; %                                                15500   0282
SVBETA [5] = .00085 ; %                                                15600   0284
SVBETA [6] = .00025 ; %                                                15700   0285
JBETASU = .00755 ; %                                                   15800   0286
SVFLAM [1] = .01246 ; %                                                15900   0287
SVFLAM [2] = .0315 ; %                                                 16000   0288
SVFLAM [3] = .1535 ; %                                                 16100   0289
SVFLAM [4] = .456 ; %                                                  16200   0291
SVFLAM [5] = 1.612 ; %                                                 16300   0292
SVFLAM [6] = 14.3 ; %                                                  16400   0293
JIMAX = 6 ; %                                                          16500   0294
L15: READ (TAPE7, FL1015, LIST3);                                      16600   0295
JSUM = 0 ; %                                                           16700   0300
GO TO SWGD1 [JNUMOPT] ; %                                              16800   0300
L20: JI = 1 ; %                                                        16900   0302
```

```
DO                                                                  17000   0303
BEGIN JSUM = JSUM + SVBETA [JI] / (SVFLAM [JI] + JALPHA)            17100   0303
END UNTIL (JI = (JI + 1)) > JIMAX;                                  17200   0305
JRHO = JALPHA * (JVITA + JSUM) / (JALPHA * JVITA + 1);              17300   0309
JREX = JRHO / (1 - JRHO) ; %                                        17400   0312
GO TO L50 ; %                                                       17500   0315
L30: SMASYMPH (JREX, JALPHA, JIMAX, SVBETA, SVFLAM, JBETASU,        17600   0329
JVITA, JSOURCE) ; %                                                 17700   0332
L50: JI = 1 ; %                                                     17800   0333
DO                                                                  17900   0333
BEGIN SVCIDOT [JI] = SVBETA [JI] * JPHI * JALPHA * (1 + JREX) / (   18000   0333
  SVFLAM [JI] + JALPHA)                                             18100   0337
END UNTIL (JI = (JI + 1)) > JIMAX;                                  18200   0338
GO TO L70 ; %                                                       18300   0341
L60: READ (TAPE7, FL1060, LIST4);                                   18400   0342
JSUM = SVCIDOT [1] ; %                                              18500   0347
JI = 2 ; %                                                          18600   0348
DO                                                                  18700   0348
BEGIN JSUM = JSUM + SVCIDOT [JI]                                    18800   0348
END UNTIL (JI = (JI + 1)) > JIMAX;                                  18900   0349
JALPHA = (JREX - JSUM / JPHI) / JVITA;                              19000   0352
L70: IF JALPHA = 0.0 THEN GO TO L71;                                19100   0355
JPERIOD = 1 / JALPHA ; %                                            19200   0357
L71: JENDOT = JPHI * JALPHA ; %                                     19300   0359
JENDOT2 = JALPHA * JENDOT ; %                                       19400   0360
WRITE (TAPE6, FL1080) ; %                                           19500   0362
WRITE (TAPE6, FL1085, LIST5) ; %                                    19600   0365
WRITE (PUNCH, FL3003, LIST5) ; %                                    19700   0369
WRITE (TAPE6, FL1090, LIST6) ; %                                    19800   0373
WRITE (TAPE6, FL1095, LIST7) ; %                                    19900   0377
GO TO SWGO2 [JNUMOPT] ; %                                           20000   0380
L100: WRITE (TAPE6, FL1100) ; %                                     20100   0382
GO TO L115 ; %                                                      20200   0386
L105: WRITE (TAPE6, FL1105) ; %                                     20300   0388
GO TO L115 ; %                                                      20400   0391
L110: WRITE (TAPE6, FL1110) ; %                                     20500   0392
L115: WRITE (TAPE6, FL1115, LIST8);                                 20600   0395
L200: READ (TAPE7, FL1200, LIST9);                                  20700   0399
JPHIZ = JPHI ; %                                                    20800   0404
JINTM = - JINTRUP ; %                                               20900   0405
IF (JNUMEXC + JINTRUP + JJC) ≠ 0 THEN GO TO L210;                   21000   0406
IF (JELTATC + JQUERY + JSTEPK) = 0 THEN GO TO L1;                   21100   0408
L210: IF (XPR = (JJC)) = 0 THEN GO TO L220;                         21200   0410
IF XPR < 0 THEN GO TO L1 ; %                                        21300   0412
```

```
JJPRINT ← JJC ; %                                                    21400  0414
L220: IF JSTEPK = 0 THEN GO TO L235;                                 21500  0414
JREX ← JREX + JSTEPK ; %                                             21600  0414
JW ← JPHI * JSTEPK ; %                                               21700  0418
JI ← 1 ; %                                                           21800  0419
DO                                                                   21900  0420
BEGIN SVCIDOT [JI] ← SVCIDOT [JI] + SVBETA [JI] * JW                 22000  0420
END UNTIL (JI ← (JI + 1)) > JIMAX;                                   22100  0422
JALPHA ← JALPHA + (1 - JBETASU) * JSTEPK / JVITA;                    22200  0425
IF JALPHA = 0.0 THEN GO TO L235;                                     22300  0429
JPERIOD ← 1 / JALPHA ; %                                             22400  0430
L235: IF JNUMEXC ≠ 0 THEN GO TO L245;                                22500  0432
IF (XPR ← (JELTATC)) > 0 THEN GO TO L250;                            22600  0433
IF XPR = 0 THEN GO TO L350 ELSE GO TO L1;                            22700  0435
L245: JNUMREX ← JNUMEXC ; %                                          22800  0436
GO TO SWGO3 [JNUMREX] ; %                                            22900  0437
L5000: JNREAD ← 2 ; %                                                23000  0439
GO TO L5050 ; %                                                      23100  0440
L5001: JNREAD ← 8 ; %                                                23200  0443
GO TO L5050 ; %                                                      23300  0443
L5002: JNREAD ← 3 ; %                                                23400  0444
GO TO L5050 ; %                                                      23500  0445
L5003: JNREAD ← 3 ; %                                                23600  0446
L5050: READ (TAPE7, FL1060, LIST10);                                 23700  0447
IF (XPR ← (JELTATC)) = 0 THEN GO TO L300;                            23800  0452
IF XPR < 0 THEN GO TO L1 ; %                                         23900  0453
L250: JBLT ← 0 ; %                                                   24000  0455
JSUMDOT ← 0 ; %                                                      24100  0455
JSL ← JS * JVITA ; %                                                 24200  0456
JTIEST ← JTLAST - .0005 * JELTATC;                                   24300  0457
JT2 ← 0.5 * JELTATC ; %                                              24400  0459
JRT2 ← 0.5 / JELTATC ; %                                             24500  0460
JRTSQR ← 1 / JELTATC * 2 ; %                                         24600  0462
JDELNM1 ← JELTATC * (JENDOT - 0.5 * JELTATC * JENDOT2);              24700  0463
JDELNM2 ← JELTATC * (JENDOT - 1.5 * JELTATC * JENDOT2);              24800  0466
JI ← 1 ; %                                                           24900  0469
DO                                                                   25000  0470
BEGIN JT2L ← JT2 * SVFLAM [JI] ; %                                   25100  0470
  SVTLIM [JI] ← 1 - JT2L ; %                                         25200  0471
  SVTLIP [JI] ← 1 + JT2L ; %                                         25300  0473
  SVRLTI [JI] ← 1 / SVTLIP [JI];                                     25400  0475
  JSUMDOT ← JSUMDOT + SVCIDOT [JI] * SVRLTI [JI];                    25500  0477
  SVBLTI [JI] ← SVBETA [JI] * SVRLTI [JI];                           25600  0479
```

```
  JBLT + JBLT + SVBLTI [JI] ; %                                    25700   0481
END UNTIL (JJ + (J1 + 1)) > JIMAX;                                 25800   0483
JBLT2 + 0.5 * JBLT ; %                                             25900   0485
JBLTVIT + JBLT2 + JVITA / JELTATC;                                 26000   0486
JCBLT + 1 - JBLT ; %                                               26100   0488
JCBLT2 + 0.5 * JCBLT ; %                                           26200   0489
JCBLT3 + 0.333333333333 * JCBLT;                                   26300   0490
JDELPHI + JELTATC * (JENDOT + JENDOT2 * JT2);                      26400   0492
JDELTAT + JELTATC ; %                                              26500   0494
L300: GO TO SWG04 [JNUMREX] ; %                                    26600   0495
L301: SVPDT [1] + SVP [1] * JDELTAT;                               26700   0497
SVPDT [2] + SVP [2] * JDELTAT ; %                                  26800   0499
SVPDT [3] + SVPDT [2] * 0.5 ; %                                    26900   0501
GO TO L350 ; %                                                     27000   0503
L302: SVPDT [1] + (((SVP [4] * JDELTAT + SVP [3]) * JDELTAT +      27100   0509
SVP [2]) * JDELTAT + SVP [1]) * JDELTAT;                           27200   0511
SVPDT [2] + ((SVP [4] * 4 * JDELTAT + 3 * SVP [3]) * JDELTAT + 2   27300   0514
* SVP [2]) * JDELTAT ; %                                           27400   0518
SVPDT [3] + (SVP [4] * 6 * JDELTAT + 3 * SVP [3]) * JDELTAT;       27500   0520
SVPDT [4] + SVP [4] * 4 * JDELTAT;                                 27600   0525
SVPDT [5] + SVP [5] * JDELTAT ; %                                  27700   0527
SVPDT [6] + SVP [6] * 2 * JDELTAT;                                 27800   0529
SVPDT [7] + SVP [7] * 3 * JDELTAT;                                 27900   0532
SVPDT [8] + SVP [8] * 4 * JDELTAT;                                 28000   0534
SVPDT [9] + SVP [6] * JDELTAT * JDELTAT;                           28100   0537
SVPDT [10] + SVPDT [7] * JDELTAT;                                  28200   0539
SVPDT [11] + SVPDT [8] * 1.5 * JDELTAT;                            28300   0541
SVPDT [12] + SVPDT [10] * .33333333333 * JDELTAT;                  28400   0544
SVPDT [13] + SVPDT [11] * .66666666667 * JDELTAT;                  28500   0546
SVPDT [14] + SVPDT [13] * .25 * JDELTAT;                           28600   0549
JT1 + 0 ; %                                                        28700   0551
GO TO L350 ; %                                                     28800   0552
L303: SVPDT [1] + SVP [1] * JDELTAT;                               28900   0557
GO TO L350 ; %                                                     29000   0559
L304: SVPDT [1] + 2 * SVP [1] * SIN (SVP [3] * JDELTAT * 0.5);     29100   0559
SVPDT [2] + SVP [2] * JDELTAT ; %                                  29200   0564
SVPDT [3] + SVPDT [2] * 0.5 ; %                                    29300   0566
SVPDT [4] + SVP [3] * JDELTAT * 0.5;                               29400   0568
L350: WRITE (TAPE6, FL1350, LIST11);                               29500   0570
WRITE (TAPE6, FL1355, LIST12) ; %                                  29600   0574
WRITE (TAPE6, FL1360) ; %                                          29700   0578
L400: JK + 1 ; %                                                   29800   0582
```

```
DO                                                                  29900   0582
BEGIN JJ ← 1 ; %                                                    30000   0582
  DO                                                                30100   0583
  BEGIN JA ← JPHI × JREX - JSUMDOT ÷ JSL;                           30200   0583
    JB ← JPHI × JCBLT2 ; %                                          30300   0586
    SVC [1] ← JBLTVIT - JREX × JCBLT2;                              30400   0587
    ASVKRTN ← 1 ; %                                                 30500   0590
    GO TO SWGO5 [JNUMREX] ; %                                       30600   0591
    L406: ASVKRTN ← 2 ; %                                           30700   0593
    JL ← 1 ; %                                                      30800   0593
    DO                                                              30900   0594
    BEGIN JTRY ← JDELK ; %                                          31000   0594
      JDELPHI ← (JA + JB × JTRY) / (SVC [1] - JCBLT3 × JTRY);       31100   0595
      GO TO SWGO6 [JNUMREX] ; %                                     31200   0598
      L407: IF (ABS (JTRY - JDELK) ≤ JEPSILO) THEN GO TO L420;      31300   0600
    END UNTIL (JL ← (JL + 1)) > 100;                                31400   0603
    WRITE (TAPE6, FL1015, LIST13);                                  31500   0605
    JIFLAG ← - 1 ; %                                                31600   0609
    GO TO L465 ; %                                                  31700   0610
    L420: JDELPHI ← (JA + JB × JDELK) / (SVC [1] - JCBLT3 ×         31800   0612
    JDELK) ; %                                                      31900   0614
    JDELK3N ← 0.333333333333 × JDELK × JDELPHI;                     32000   0615
    JTHERM ← JTHERM + (JPHI + JDELPHI × 0.5) × JDELTAT;             32100   0617
    JSUMDOT ← 0 ; %                                                 32200   0621
    JX ← JDELK × JPHI + (1 + JREX + 0.666666666666 × JDELK) ×       32300   0621
    JDELPHI ; %                                                     32400   0624
    JI ← 1 ; %                                                      32500   0626
    DO                                                              32600   0626
    BEGIN SVCIDOT [JI] ← (SVBETA [JI] × (JX + SVTLIP [JI] ×         32700   0626
      JDELK3N) + SVCIDOT [JI] × SVTLIM [JI]) × SVRLTI [JI];         32800   0628
      JSUMDOT ← JSUMDOT + SVCIDOT [JI] × SVRLTI [JI];               32900   0632
    END UNTIL (JI ← (JI + 1)) > JIMAX;                              33000   0634
    JREX ← JREX + JDELK ; %                                         33100   0636
    JPHI ← JPHI + JDELPHI ; %                                       33200   0638
    JDELNM2 ← JDELNM1 ; %                                           33300   0640
    JDELNM1 ← JDELPHI ; %                                           33400   0641
    JENDOT ← (3 × JDELNM1 - JDELNM2) × JRT2;                        33500   0642
    JENDOT2 ← (JDELNM1 - JDELNM2) × JRTSQR;                         33600   0644
    JALPHA ← JENDOT / JPHI ; %                                      33700   0646
    JDELPHI ← 2 × JDELNM1 - JDELNM2;                                33800   0648
    JT ← JT + JDELTAT ; %                                           33900   0650
    JT1 ← JT1 + JDELTAT ; %                                         34000   0652
    IF (JT < JTTEST) THEN GO TO L440;                               34100   0653
```

```
      JIFLAG ← = 1 ; %                                                    34200  0654
      GO TO L465 ; %                                                      34300  0655
      L440: IF (XPR ← (JINTRUP)) = 0 THEN GO TO L460;                     34400  0660
      IF XPR < 0 THEN GO TO L446 ; %                                      34500  0661
      IF (SVTESTO [JINTRUP] ≥ JQUERY) THEN GO TO L459 ELSE GO TO          34600  0663
      L460 ; %                                                            34700  0664
      L446: IF (SVTESTO [JINTM] > JQUERY) THEN GO TO L460;                34800  0665
      L459: JIFLAG ← 1 ; %                                                34900  0666
      GO TO L465 ; %                                                      35000  0667
      L460:                                                               35100  0668
    END UNTIL (JJ ← (JJ + 1)) > JJPRINT;                                  35200  0669
    L465: JPERIOD ← 1 / JALPHA ; %                                        35300  0671
    WRITE (TAPE6, FL1480, LIST14);                                        35400  0673
    JI ← 1 ; %                                                            35500  0677
    DO                                                                    35600  0678
    BEGIN SVC [JI] ← ((1 + JREX) × SVBETA [JI] × JPHI - SVCIDOT [         35700  0678
      JI]) / (JVITA × SVFLAM [JI])                                        35800  0681
    END UNTIL (JI ← (JI + 1)) > JIMAX;                                    35900  0682
    WRITE (TAPE6, FL3001, LIST15);                                        36000  0685
    WRITE (PUNCH, FL3002, LIST16);                                        36100  0689
    IF JIFLAG ≠ 0 THEN GO TO L480;                                        36200  0693
  END UNTIL (JK ← (JK + 1)) > 1000;                                       36300  0694
  L480: IF (XPR ← (JIFLAG)) > 0 THEN GO TO L490;                          36400  0696
  IF XPR = 0 THEN GO TO L400 ; %                                          36500  0698
  JIFLAG ← 0 ; %                                                          36600  0700
  GO TO L1 ; %                                                            36700  0700
  L490: JIFLAG ← 0 ; %                                                    36800  0701
  GO TO L200 ; %                                                          36900  0702
  L2100: JDELK1 ← SVPDT [1] + SVPDT [2] × JPHI;                           37000  0703
  L2101: JDELK ← JDELK1 + SVPDT [3] × JDELPHI;                            37100  0706
  GO TO L2000 ; %                                                         37200  0709
  L2102: JDELK1 ← ((SVPDT [4] × JT1 + SVPDT [3]) × JT1 + SVPDT [2]        37300  0709
  ) × JT1 + SVPDT [1] ; %                                                 37400  0712
  JDK1 ← ((SVPDT [8] × JTHERM + SVPDT [7]) × JTHERM + SVPDT [6]) ×        37500  0714
  JTHERM + SVPDT [5] ; %                                                  37600  0718
  JDK2 ← (SVPDT [11] × JTHERM + SVPDT [10]) × JTHERM + SVPDT [9];         37700  0720
  JDK3 ← SVPDT [13] × JTHERM + SVPDT [12];                                37800  0724
  L2103: JX ← JPHI + 0,5 × JDELPHI;                                       37900  0726
  JDELK ← JDELK1 + JX × (((JX × SVPDT [14] + JDK3) × JX + JDK2) ×         38000  0729
  JX + JDK1) ; %                                                          38100  0732
```

```
   GO TO L2000 ; %                                                    38200    0734
   L2200: JX = JDELPHI / JPHI ; %                                     38300    0736
   IF (ABS (JX) < .01) THEN GO TO L2210;                              38400    0737
   JX = SVP [2] * LN (1 + JX) ; %                                     38500    0739
   GO TO L2215 ; %                                                    38600    0741
   L2210: JX = JX * (1 + JX * (= .5 + JX * (.333333333333 = .25 *     38700    0743
   JX))) * SVP [2] ; %                                                38800    0745
   L2215: JDELK = (ABS (SVPDT [1]) * SIGN (SVP [3] = JT)) + JX;       38900    0748
   L2104: JDELK1 = SVPDT [1] * COS (SVP [3] * JT + SVPDT [4]) +       39000    0754
   SVPDT [2] * (JPHI = JPHIZ) ; %                                     39100    0757
   L2105: JDELK = JDELK1 + SVPDT [3] * JDELPHI;                       39200    0759
   L2000: GO TO ASSIGN (ASVKRTN) ; %                                  39300    0762
   L2001: WRITE (TAPE6, FL1) ; %                                      39400    0764
   LEND:                                                              39500    0767
END ; %                                                               39600    0768
                                                    0004 IS 0798 LONG, NEXT SEG 0002
COMMENT  INITIALIZING BLOCK ; %                                       39700    0035
XPR = Q = K = 0 ; %                                                   39800    0035
SENSL [1] = FALSE ; %                                                 39900    0036
SENSL [2] = FALSE ; %                                                 40000    0038
SENSL [3] = FALSE ; %                                                 40100    0039
SENSL [4] = FALSE ; %                                                 40200    0040
MAINPRO END.                                                          40300    0041
```

```
COMMENT   R O O R O P ;                                                    0000
 BEGIN FILE TAPE6 1 (2, 15) ; %                                    100    0000
                                                  START OF SEGMENT ********** 0002
 FILE TAPE7 0 (2, 15) ; %                                          200    0005
 FILE XXXXXX 2 (2, 15) ; %                                         300    0010
 SWITCH FILE FILESW = XXXXXX, XXXXXX, XXXXXX, XXXXXX, XXXXXX,      400    0015
 XXXXXX, TAPE6, TAPE7 ; %                                          500    0023
 BOOLEAN ARRAY SENSL [0 : 4], SENSW [0 : 6];                       600    0027
 REAL ARRAY SVOC [0 : 20], SVBETA [0 : 20], SVS [0 : 21], SVBSO [0 700    0031
 : 21, 0 : 20], SVBOD [0 : 21, 0 : 20];                           800    0037
 INTEGER JN ; %                                                    900    0041
 INTEGER K ; %                                                     1000   0041
 REAL JPNL ; %                                                     1100   0041
 REAL Q, XPR ; %                                                   1200   0041
 LABEL FINIS ; %                                                   1300   0041
 FORMAT F (/ / / / / / "STOP / PAUSE  NO. ", I5);                  1400   0041
                                                  START OF SEGMENT ********** 0003
                                                  0003 IS 0013 LONG, NEXT SEG 0002
 REAL PROCEDURE INT (ARG1) ; %                                     1500   0041
 VALUE ARG1 ; %                                                    1600   0041
 REAL ARG1 ; %                                                     1700   0041
 INT = SIGN (ARG1) × ENTIER (ABS (ARG1));                          1800   0041
 REAL PROCEDURE TANH (ARG1) ; %                                    1900   0049
 VALUE ARG1 ; %                                                    2000   0049
 REAL ARG1 ; %                                                     2100   0049
 TANH = ((Q = EXP (ARG1 × 2)) - 1) / (Q + 1);                      2200   0049
 REAL PROCEDURE MAX (ARG1, ARG2) ; %                               2300   0055
 VALUE ARG1, ARG2 ; %                                              2400   0055
 REAL ARG1, ARG2 ; %                                               2500   0055
 MAX = IF ARG1 ≥ ARG2 THEN ARG1 ELSE ARG2;                         2600   0055
 REAL PROCEDURE MIN (ARG1, ARG2) ; %                               2700   0060
 VALUE ARG1, ARG2 ; %                                              2800   0060
 REAL ARG1, ARG2 ; %                                               2900   0060
 MIN = IF ARG1 ≤ ARG2 THEN ARG1 ELSE ARG2;                         3000   0060
 REAL PROCEDURE DIM (ARG1, ARG2) ; %                               3100   0065
 VALUE ARG1, ARG2 ; %                                              3200   0065
 REAL ARG1, ARG2 ; %                                               3300   0065
 DIM = MAX (ARG1 - ARG2, 0) ; %                                    3400   0065
 PROCEDURE ERROR (ARG1) ; %                                        3500   0069
 VALUE ARG1 ; %                                                    3600   0069
 REAL ARG1 ; %                                                     3700   0069
 BEGIN WRITE (TAPE6, F, ARG1) ; %                                  3800   0069
   GO TO FINIS                                                     3900   0077
 END ; %                                                           4000   0078
```

```
PROCEDURE SRROUTIN (JR) ; %                                          4100    0080
VALUE JR ; %                                                         4200    0080
REAL JR ; %                                                          4300    0080
BEGIN INTEGER JNA, JK, JJ, JIT, JI;                                  4400    0080
                                                  START OF SEGMENT ********** 0004
  REAL JRHO, JRHOOFS, JSO, JDEV, JPREDEV, JU, JV, JRHOOOT, JDELTA;   4500    0000
  LABEL L7, L139, L17, L19, L23, L20, L26, L250, L134;               4600    0000
  JNA ← JN + 1 ; %                                                   4700    0000
  JRHO ← JR ; %                                                      4800    0001
  SVS [1] ← 0 ; %                                                    4900    0002
  JK ← 2 ; %                                                         5000    0003
  DO                                                                 5100    0004
  BEGIN SVS [JK] ← = (SVDC [JK] + SVDC [JK - 1]) / 2                 5200    0004
  END UNTIL (JK ← (JK + 1)) > JN ; %                                 5300    0006
  SVS [JNA] ← = (1 / JPNL + SVDC [JN]) / 2;                          5400    0009
  JJ ← 1 ; %                                                         5500    0013
  DO                                                                 5600    0013
  BEGIN JIT ← 1 ; %                                                  5700    0013
    L7: JRHOOFS ← 0 ; %                                              5800    0014
    JI ← 1 ; %                                                       5900    0015
    DO                                                               6000    0016
    BEGIN JSO ← SVS [JJ] + SVDC [JI];                                6100    0016
      SVBSO [JJ, JI] ← SVBETA [JI] / JSO;                            6200    0018
      SVBOO [JJ, JI] ← SVBSO [JJ, JI] / JSO;                         6300    0021
      JRHOOFS ← JRHOOFS + SVBSO [JJ, JI];                            6400    0024
    END UNTIL (JI ← (JI + 1)) > JN;                                  6500    0026
    JRHOOFS ← SVS [JJ] × (JPNL + JRHOOFS) / (1 + JPNL × SVS [JJ]);   6600    0029
    JDEV ← JRHOOFS - JRHO ; %                                        6700    0032
    IF NOT (JIT > 1) THEN GO TO L139;                                6800    0034
    IF (XPR ← (JDEV × JPREDEV)) = 0 THEN GO TO L134;                 6900    0035
    IF XPR < 0 THEN GO TO L139 ; %                                   7000    0037
    IF NOT (ABS (JDEV) < ABS (JPREDEV)) THEN GO TO L134;             7100    0038
    L139: IF NOT (JIT < 100) THEN GO TO L134;                        7200    0040
    JPREDEV ← JDEV ; %                                               7300    0042
    JIT ← JIT + 1 ; %                                                7400    0043
    JU ← 0 ; %                                                       7500    0044
    JV ← 0 ; %                                                       7600    0045
    JI ← 1 ; %                                                       7700    0045
    DO                                                               7800    0046
    BEGIN JU ← JU + SVBSO [JJ, JI];                                  7900    0046
      JV ← JV + SVBOO [JJ, JI]                                       8000    0048
    END UNTIL (JI ← (JI + 1)) > JN;                                  8100    0050
    JRHOOOT ← ((JPNL + JU) / (1 + JPNL × SVS [JJ]) - SVS [JJ] × JV   8200    0053
```

```
    ) / (1 + JPNL * SVS [JJ]) ; %                                    8300    0056
    JDELTA + = JDEV / JRHODOT ; %                                    8400    0059
    IF (JJ ≤ 1) THEN GO TO L17 ; %                                   8500    0060
    IF (JJ = JN ≥ 1) THEN GO TO L20 ELSE GO TO L19;                  8600    0062
    L17: IF (SVS [JJ] + JDELTA + SVDC [1]) > 0 THEN GO TO L250;      8700    0064
    JDELTA + JDELTA / 10 ; %                                         8800    0067
    GO TO L17 ; %                                                    8900    0069
    L19: IF (SVS [JJ] + JDELTA + SVDC [JJ]) > 0 THEN GO TO L23;      9000    0069
    JDELTA + JDELTA / 10 ; %                                         9100    0072
    GO TO L19 ; %                                                    9200    0074
    L23: IF (= SVDC [JJ - 1] - SVS [JJ] > JDELTA) THEN GO TO L250;   9300    0074
    JDELTA + JDELTA / 10 ; %                                         9400    0078
    GO TO L23 ; %                                                    9500    0079
    L20: IF (SVS [JJ] + JDELTA + 1 / JPNL) > 0 THEN GO TO L26;       9600    0079
    JDELTA + JDELTA / 10 ; %                                         9700    0083
    GO TO L20 ; %                                                    9800    0084
    L26: IF (= SVDC [JN] - SVS [JJ] > JDELTA) THEN GO TO L250;       9900    0084
    JDELTA + JDELTA / 10 ; %                                        10000    0087
    GO TO L26 ; %                                                   10100    0088
    L250: SVS [JJ] + SVS [JJ] + JDELTA;                             10200    0089
    GO TO L7 ; %                                                    10300    0092
    L134:                                                           10400    0092
  END UNTIL (JJ + (JJ + 1)) > JNA;                                  10500    0093
END ; %                                                             10600    0095

                                                  0004 IS 0101 LONG, NEXT SEG 0002
PROCEDURE MAINPRO ; %                                               10700    0080
BEGIN INTEGER ARRAY SVIDENT [0 : 9];                                10800    0080

                                                  START OF SEGMENT ********** 0005
  REAL ARRAY SVC [0 : 20], SVA [0 : 21], SVAA [0 : 21], SVAB [0 :   10900    0002
  21] ; %                                                           11000    0008
  INTEGER DX1 ; %                                                   11100    0010
  INTEGER JNA, JJJ, JII, JJ, JL ; %                                 11200    0010
  REAL JTDROP, JEXK, JPSTART, JEFK, JRHO, JSUMA, JSUMAA, JSUMD,     11300    0010
  JDELTAT, JDELT, JFRACT, JX, JT, JPH, JPI, JFACTOR, JERROR;        11400    0010
  COMMENT   THE FOLLOWING SUBROUTINES ARE REQUIRED:                 11500    0010
          SRROOTIN ; %                                              11600    0010
  FORMAT FL2 (I12, R12, 5 / (6 R12, 5)), FL1 (9 A6), FL3 (4 R20,    11700    0010

                                                  START OF SEGMENT ********** 0006
  10), FL4 (" ", 9 A6 / / " PNL =", R20, 10, "  EXCESS K =", R20,  11800    0010
  10, "  REACTIVITY =", R20, 10 / / " DROP TIME =", R20, 10,        11900    0010
  " SECONDS, STARTING FLUX =", R20, 10 / /                          12000    0010
  "                LAMBDA                 DELTA  STARTING FREQ CORR," /   12100    0010
  / (3 R20, 10)), FL8 (/ " TIME DISPLACEMENT", R20, 10, " SECONDS"  12200    0010
  / /                                                               12300    0010
```

```
"                    ROOT                    A(IDEAL)              "          12400   0010
, "     A(REAL)         A(IDEAL)CORRECTED" / / (4 E25. 10)), FL21              12500   0010
(/ " EQUIVALENT DROP", R20. 9, "  SECONDS OR", R15. 9,                         12600   0010
"  DROP TIMES AFTER TIME ZERO"), FL9 (/                                       12700   0010
"          TIME AFTER DROP                  REAL FLUX     IDEAL FLU"        12800   0010
, "X(DISPLACED)           RELATIVE ERROR"), FL12 (R25. 2, 3 R25.             12900   0010
10) ; %                                                                      13000   0010
                                                                  0006 IS 0142 LONG, NEXT SEG 0005
LIST LIST1 (JN, JPNL, FOR DX1 ← 1 STEP 1 UNTIL JN DO SVBETA (DX1             13100   0010
), FOR DX1 ← 1 STEP 1 UNTIL JN DO SVDC (DX1));                               13200   0015
LIST LIST2 (FOR DX1 ← 1 STEP 1 UNTIL 9 DO SVIDENT (DX1));                    13300   0027
LIST LIST3 (JTDROP, JEXK, JPSTART, SVC (1));                                 13400   0037
LIST LIST4 (FOR DX1 ← 2 STEP 1 UNTIL JN DO SVC (DX1));                       13500   0047
LIST LIST5 (FOR DX1 ← 1 STEP 1 UNTIL 9 DO SVIDENT (DX1), JPNL,               13600   0057
JEXK, JRHO, JTDROP, JPSTART, FOR DX1 ← 1 STEP 1 UNTIL JN DO (                13700   0065
SVDC (DX1), SVBETA (DX1), SVC (DX1)));                                       13800   0071
LIST LIST6 (JDELTAT, FOR DX1 ← 1 STEP 1 UNTIL JNA DO (SVS (DX1),             13900   0081
SVA (DX1), SVAA (DX1), SVAB (DX1)));                                         14000   0086
LIST LIST7 (JDELT, JFRACT) ; %                                               14100   0097
LIST LIST8 (JT, JPN, JPI, JERROR);                                           14200   0105
LABEL L20, L14, L15, LEND ; %                                                14300   0115
READ (TAPE7, FL2, LIST1) ; %                                                 14400   0115
L20: READ (TAPE7, FL1, LIST2) (LEND);                                        14500   0119
L14: READ (TAPE7, FL3, LIST3) ; %                                            14600   0124
IF JTDROP ≤ 0 THEN GO TO L20 ; %                                             14700   0128
READ (TAPE7, FL3, LIST4) ; %                                                 14800   0129
JEFK ← JEXK + 1 ; %                                                          14900   0133
JRHO ← JEXK / JEFK ; %                                                       15000   0134
WRITE (TAPE6 (PAGE)) ; %                                                     15100   0135
WRITE (TAPE6, FL4, LIST5) ; %                                                15200   0139
SRHODTIN (JRHO) ; %                                                          15300   0142
JNA ← JN + 1 ; %                                                             15400   0143
JJJ ← 1 ; %                                                                  15500   0144
DO                                                                           15600   0145
BEGIN JSUMA ← 0 ; %                                                          15700   0145
  JSUMAA ← 0 ; %                                                             15800   0146
  JSUMD ← 0 ; %                                                              15900   0147
  JII ← 1 ; %                                                                16000   0147
  DO                                                                         16100   0148
  BEGIN JSUMA ← JSUMA + SVBSD (JJJ, JII);                                    16200   0148
    JSUMAA ← JSUMAA + SVC (JII) × SVDC (JII) / (SVS (JJJ) + SVDC             16300   0150
    (JII)) ; %                                                               16400   0152
    JSUMD ← JSUMD + SVDC (JII) × SVBDD (JJJ, JII)                            16500   0154
```

```
    END UNTIL (JII + (JII + 1)) > JN;                                    16600   0156
    JSUMD + JSUMD + JEFK / JPNL + 1;                                     16700   0159
    SVA [JJJ] + (1 + JSUMA / JPNL) / JSUMD;                              16800   0162
    SVAA [JJJ] + (JPSTART + JSUMAA) / JSUMD                              16900   0164
  END UNTIL (JJJ + (JJJ + 1)) > JNA;                                     17000   0166
  JDELTAT + LN (SVAA [1] / SVA [1]) / SVS [1];                           17100   0169
  JJ + 1 ; %                                                             17200   0172
  DO                                                                     17300   0173
  BEGIN SVAB [JJJ] + SVA [JJ] + EXP (SVS [JJ] + JDELTAT)                 17400   0173
  END UNTIL (JJ + (JJ + 1)) > JNA;                                       17500   0174
  WRITE (TAPE6, FL8, LIST6) ; %                                          17600   0178
  JDELT + JTDROP - JDELTAT ; %                                           17700   0182
  JFRACT + JDELT / JTDROP ; %                                            17800   0183
  WRITE (TAPE6, FL21, LIST7) ; %                                         17900   0184
  WRITE (TAPE6, FL9) ; %                                                 18000   0188
  JX + .01 ; %                                                           18100   0192
  L15: JL + 1 ; %                                                        18200   0192
  DO                                                                     18300   0193
  BEGIN JT + JX * JL ; %                                                 18400   0193
    JPR + 0 ; %                                                          18500   0195
    JPI + 0 ; %                                                          18600   0195
    JJ + 1 ; %                                                           18700   0196
    DO                                                                   18800   0197
    BEGIN JFACTOR + EXP (SVS [JJ] + JT);                                 18900   0197
      JPR + JPR + SVAA [JJ] + JFACTOR;                                   19000   0199
      JPI + JPI + SVAB [JJ] + JFACTOR                                    19100   0201
    END UNTIL (JJ + (JJ + 1)) > JNA;                                     19200   0202
    JERROR + (JPR - JPI) / JPR ; %                                       19300   0205
    WRITE (TAPE6, FL12, LIST8) ; %                                       19400   0207
  END UNTIL (JL + (JL + 1)) > 9 ; %                                      19500   0211
  IF (JX ≥ 100) THEN GO TO L14 ; %                                       19600   0213
  JX + JX + 10 ; %                                                       19700   0214
  GO TO L15 ; %                                                          19800   0215
  LEND:                                                                  19900   0222
END ; %                                                                  20000   0222
                                                0005 IS 0234 LONG, NEXT SEG 0002
COMMENT  INITIALIZING BLOCK ; %                                          20100   0080
XPR + Q + K + 0 ; %                                                      20200   0080
SENSW [1] + FALSE ; %                                                    20300   0081
SENSW [2] + FALSE ; %                                                    20400   0083
SENSW [3] + FALSE ; %                                                    20500   0084
SENSW [4] + FALSE ; %                                                    20600   0085
SENSW [5] + FALSE ; %                                                    20700   0086
SENSW [6] + FALSE ; %                                                    20800   0088
```

```
SENSL [1] ← FALSE ) %                                    20900    0089
SENSL [2] ← FALSE ) %                                    21000    0090
SENSL [3] ← FALSE ) %                                    21100    0091
SENSL [4] ← FALSE ) %                                    21200    0093
MAINPRO ) %                                              21300    0094
FINIS: END.                                              21400    0094
```

APPENDIX D

PROCEDURE DATARED

The purpose of this data reduction procedure for the B-5500 computer is to process the data and coded information which is collected at the reactor before commencing the least squares curve fitting. The coded information consists of a test number, multiscaler sequence, and multiscaler dwell-time code. This information is manually punched into the leader of the paper tape prior to data readout onto the tape from a multiscaler. The data which follow the coded information are simply the contents of each memory location of the multiscaler, punched serially. An IBM tape-card printing punch has been provided with a wired-board program to convert this information into punched cards since the B-5500 is not provided with a paper tape input reader. These punched cards form a portion of the input to DATARED. The other input required is the test number, total number of tapes for which information is being processed, a code indicating whether the number of channels from the multiscaler was other than 400, the switching time between multiscalers, the detection and recording system dead time, the dwell time to be associated with a special non-standard dwell-time code, and the time after experiment start to be associated with the first piece of data. In addition, if the code indicates other than 400 channels of data, the number of channels must be provided. The final input is a reduction factor for each time code. This factor tells the procedure how many channels of data to combine

into a single data point for economy in analysis. This factor gives the fine control on total number of data points which must be processed.

In the program, data are corrected for dead time losses and combined with other data points as determined by the reduction factor described above. A value of the standard deviation, σ, is obtained by taking the square root of the corrected counts for the data point. Using the dwell time per channel, elapsed time prior to the start of the multiscaler being considered, and multiscaler switching time, an interval start and stop time are calculated for each combined data point. The end result consists of one punched card per final data point and includes interval start and stop time, total counts over the interval, and the standard deviation.

Procedure DATARED is listed on the following pages.

```
COMMENT   D A T A R E D ;                                                    0000
 BEGIN FILE IN TAPE7 0 (2, 15) ; %                                    100    0000
                                                  START OF SEGMENT ********** 0002
 FILE OUT PUNCH 0 (2, 10) ; %                                         200    0005
 FILE OUT TAPE6 1 (2, 15) ; %                                         300    0010
 COMMENT    DATARED ; %                                               400    0015
 INTEGER I, CHAND, NOTEST, TESTNO, TOTAPE, CODE, FIN, DIG, TCODE,     500    0015
 PTAPE, INIT, START, NOTHIN ; %                                       600    0015
 INTEGER J8, JMODE, K ; %                                             700    0015
 REAL ELAP ; %                                                        800    0015
 REAL TCHAN, EXTCH, DEL, SUM, TAU ; %                                 900    0015
 INTEGER ARRAY REDFACT [0 : 20] ; %                                  1000    0015
 REAL ARRAY SVY [0 : 1022], SVSIGMA [0 : 1022], SVT [0 : 1022], SVU  1100    0017
 [0 : 1022], TZERO [0 : 1022] ; %                                    1200    0023
 LABEL LSTART, L30, L1, L3, L4, L5, L6, L7, L8, L9, L10, L11, L12,   1300    0027
 L13, L14, L15, L16, LCHEN, L17, L18, L19, L41, L43, L45, LEND,      1400    0027
 DUMMY, L181, L182, L183, L184 ; %                                   1500    0027
 FORMAT FL3001 (3 I1), FL1 (R10. 4, X4, R10. 4, X4, R14. 6, X4,      1600    0027
                                                  START OF SEGMENT ********** 0003
 R14. 6), FL2 (R12. 4), FL9 (20 I3), FL3002 (X19, 5 (X4, I6)),       1700    0027
 FL3003 (4 I4, 4 R13. 3), FL3004 (I5), FL3005 (/                     1800    0027
 "ERROR1=DISCREPANCY BETWEEN INPUT TEST NUMBER AND TEST              1900    0027
NUMBER ON PAPER TAPE"), FL3006 (                                     2000    0027
 "ERROR2=PAPER TAPE DOES NOT APPEAR TO BE CORRECTLY SEQUENCED"),     2100    0027
 FL3007 (/ "ERROR3=PAPER TAPE TIME CODE IS INVALID ZERO"), FL3008 (  2200    0027
 / "ERROR4=PAPER TAPE TIME CODE IS INVALID"), FL5001 (/              2300    0027
 "          T              Y            SIGMA           "), FL6001 (/ 2400    0027
 "  ", 3 E14. 6) ; %                                                 2500    0027
                                                  0003 IS 0113 LONG, NEXT SEG 0002
 LIST LIST1 (TZERO [I], SVU [I], SVY [I], SVSIGMA [I]) ;             2600    0027
 LIST LIST5 (TESTNO, DIG, TCODE) ; %                                 2700    0038
 LIST LIST6 (FOR I ← INIT STEP 1 UNTIL FIN DO SVY [I]) ;             2800    0047
 LIST LIST7 (NOTEST, TOTAPE, CODE, JMODE, DEL, EXTCH, TAU, SVU [0])  2900    0057
  ; %                                                                3000    0068
 LIST LIST8 (CHAND) ; %                                              3100    0072
 COMMENT   REDFACT MUST BE ABLE TO DIVIDE INTO CHAND WITHOUT REMAINDER; 3200    0078
 LIST LIST9 (FOR I ← 1 STEP 1 UNTIL 20 DO REDFACT [I]) ;             3300    0078
 LIST LIST41 (SVT [I], SVY [I], SVSIGMA [I]) ;                       3400    0088
 LSTART: READ (TAPE7, FL3003, LIST7) [LEND] ;                        3500    0098
 READ (TAPE7, FL9, LIST9) ; %                                        3600    0103
 PTAPE ← 1 ; %                                                       3700    0107
 INIT ← 0 ; %                                                        3800    0107
 FIN ← 0 ; %                                                         3900    0108
```

```
READ (TAPE7, FL2, ELAP) ; %                                    4000   0109
IF CODE = 0 THEN GO TO L30 ELSE GO TO L1;                      4100   0121
L30: CHANO ← 400 ; %                                           4200   0123
GO TO L3 ; %                                                   4300   0123
L1: READ (TAPE7, FL3004, LIST8) ; %                            4400   0126
L3: READ (TAPE7, FL3001, LIST5) ; %                            4500   0129
IF NOTEST = TESTNO THEN GO TO L5 ELSE GO TO L4;                4600   0133
L4: WRITE (TAPE6, FL3005) ; %                                  4700   0134
GO TO LEND ; %                                                 4800   0138
L5: IF DIG = PTAPE THEN GO TO L7 ELSE GO TO L6;                4900   0141
L6: WRITE (TAPE6, FL3006) ; %                                  5000   0142
GO TO LEND ; %                                                 5100   0146
L7: IF TCODE = 0 THEN GO TO L8 ELSE GO TO L9;                  5200   0147
L8: WRITE (TAPE6, FL3007) ; %                                  5300   0148
GO TO LEND ; %                                                 5400   0152
L9: IF TCODE = 1 THEN GO TO L10 ELSE GO TO L11;                5500   0153
L10: TCHAN ← 0.01 ; %                                          5600   0154
GO TO LCHEN ; %                                                5700   0155
L11: IF TCODE = 2 THEN GO TO L12 ELSE GO TO L13;               5800   0158
L12: TCHAN ← 0.1 ; %                                           5900   0159
GO TO LCHEN ; %                                                6000   0160
L13: IF TCODE = 3 THEN GO TO L14 ELSE GO TO L15;               6100   0163
L14: TCHAN ← 1.0 ; %                                           6200   0164
GO TO LCHEN ; %                                                6300   0165
L15: IF TCODE = 4 THEN GO TO L16 ELSE GO TO L17;               6400   0168
L16: TCHAN ← 10.0 ; %                                          6500   0169
GO TO LCHEN ; %                                                6600   0170
L17: IF TCODE = 5 THEN GO TO L18 ELSE GO TO L181;              6700   0173
L18: TCHAN ← 100.00 ; %                                        6800   0174
GO TO LCHEN ; %                                                6900   0175
L181: IF TCODE = 6 THEN GO TO L182 ELSE GO TO L183;            7000   0178
L182: TCHAN ← 1100.0 ; %                                       7100   0179
GO TO LCHEN ; %                                                7200   0180
L183: IF TCODE = 7 THEN GO TO L184 ELSE GO TO L19;             7300   0183
L184: TCHAN ← EXTCH ; %                                        7400   0184
GO TO LCHEN ; %                                                7500   0185
L19: WRITE (TAPE6, FL3008) ; %                                 7600   0186
GO TO LEND ; %                                                 7700   0190
LCHEN: INIT ← FIN ← 1 ; %                                      7800   0191
START ← INIT ; %                                               7900   0192
FIN ← INIT + CHANO - 1 ; %                                     8000   0193
READ (TAPE7, FL3002, LIST6) ; %                                8100   0194
I ← INIT ; %                                                   8200   0198
```

```
FIN ← INIT + CHANO / REDFACT [TCODE] - 1;                          8300   0199
DO                                                                 8400   0202
BEGIN SVU [I] ← ELAP + (I - START + 1) × REDFACT [TCODE] × TCHAN;   8500   0202
  TZERO [I] ← SVU [I] - REDFACT [TCODE] × TCHAN;                   8600   0206
  SUM ← 0 ; %                                                      8700   0208
  FOR K ← INIT STEP 1 UNTIL (INIT + REDFACT [TCODE] - 1) DO        8800   0209
  BEGIN SUM ← SUM + SVY [K] / (1 - (SVY [K] / TCHAN) × TAU)        8900   0214
  END ; %                                                          9000   0216
  SVY [I] ← SUM ; %                                                9100   0220
  SVSIGMA [I] ← SQRT (SVY [I]) ; %                                 9200   0221
  INIT ← INIT + REDFACT [TCODE] ; %                                9300   0223
END UNTIL (I ← (I + 1)) > FIN ; %                                  9400   0224
FOR I ← START STEP 1 UNTIL FIN DO                                  9500   0227
BEGIN WRITE (TAPE6, FL1, LIST1) ; %                                9600   0228
  WRITE (PUNCH, FL1, LIST1)                                        9700   0231
END ; %                                                            9800   0235
IF PTAPE = TOTAPE THEN GO TO L45 ELSE GO TO L41;                   9900   0237
L41: PTAPE ← PTAPE + 1 ; %                                         10000  0239
ELAP ← ELAP + CHANO × TCHAN + DEL;                                 10100  0241
L43: GO TO L1 ; %                                                  10200  0243
L45: READ (TAPE7, FL3004, NOTHIN) [LEND];                          10300  0244
IF NOTHIN = 0 THEN GO TO LSTART ; %                                10400  0256
LEND: END.                                                         10500  0257
```

APPENDIX E

VARIABLE METRIC MINIMIZATION WITH INTEGRAL FUNCTION PROCEDURE

Variable Metric Minimization[33] is a method for determining numerically the local minima of differentiable functions of several variables. In the process of determining these minima, there is generated a matrix which characterizes the behavior of the function about the minimum.

Variable Metric Minimization was originally coded in FORTRAN at the Argonne National Laboratory and subsequently translated into ALGOL at Georgia Tech for the B-5500 computer. It is a general purpose method in that any differentiable function may be minimized. A separate procedure must be provided in which the value of the particular function and its derivatives with respect to each variable are evaluated at a given trial point. In the method employed in this work, weighted least squares was the goodness-of-fit criterion used. Thus, an expression was developed which incorporated this method of curve fitting into the required function to be minimized. The Variable Metric Method is explained in detail in Reference 33; only the special integral function procedure will be described herein.

If flux decay data were taken by means of instantaneous samplings at progressively longer times after shutdown, then an appropriate weighted least squares function to be minimized would be

$$F = \sum_{j=1}^{B} \left(\frac{y_j - \sum_{i=1}^{M} a_i' e^{-\lambda_i' t_j}}{\sigma_j} \right)^2 \qquad (6)$$

where B = number of data points

M = number of delayed neutron groups

y_j = experimentally observed neutron count at the j^{th} time, t_j

σ_j = standard deviation of j^{th} data point (the weighting factor)

t_j = time associated with the j^{th} data point

a_i' = group abundance term, initial count rate attributable to i^{th} group

λ_i' = group decay constant uncorrected for sub-critical multiplication

In fact, since the data are not observed instantaneously, but over an interval, Δt_j, starting at t_j, the function which in practice must be minimized is

$$F = \sum_{j=1}^{B} \left(\frac{y_j - \sum_{i=1}^{M} \int_{t_j}^{t_j+\Delta t_j} a_i' e^{-\lambda_i' t} dt}{\sigma_j} \right)^2$$

When the integration is performed, the result is

$$F = \sum_{j=1}^{B} \left(\frac{y_j - \sum_{i=1}^{M} \frac{a_i'}{\lambda_i'} \left[e^{-\lambda_i' t_j} - e^{-\lambda_i' (t_j+\Delta t_j)} \right]}{\sigma_j} \right)^2$$

and the gradient vector is obtained from

$$\frac{\partial F}{\partial a_i'} = \sum_{j=1}^{B} -2\left[y_j - \sum_{i=1}^{M} \frac{a_i'}{\lambda_i'}\left(e^{-\lambda_i' t_j} - e^{-\lambda_i'(t_j+\Delta t_j)}\right)\right] \times$$

$$\left[\frac{e^{-\lambda_i' t_j}}{\sigma_j^2 \lambda_i'}\left(1 - e^{-\lambda_i' \Delta t_j}\right)\right]$$

and

$$\frac{\partial F}{\partial \lambda_i'} = \sum_{j=1}^{B} -2\left[y_j - \sum_{i=1}^{M} \frac{a_i'}{\lambda_i'}\left(e^{-\lambda_i' t_j} - e^{-\lambda_i'(t_j+\Delta t_j)}\right)\right] \times$$

$$\left[\frac{a_i e^{-\lambda_i' t_j}}{\lambda_i'}\left(\Delta t\, e^{-\lambda_i' \Delta t_j} + (e^{-\lambda_i' \Delta t_j} - 1)(t_j + \frac{1}{\lambda_i'})\right)\right]$$

Thus, each time the main Variable Metric Minimization Program calls this Integral Function Procedure, it passes to it the one-dimensional array representing the present estimate of the parameters a_i' and λ_i' and receives back the value of the least squares function calculated using those estimates, plus the one-dimensional array representing the gradient vector at the point defined by the parameter estimates. This gradient vector is then used by the main program in conjunction with the error matrix, which carries the most current information on the uncertainty of the present parameter estimates, to obtain a correction value which may be added to the parameters before their next trial. This iteration procedure continues until the predicted improvement for the next cycle is less than an input-specified con-

vergence criterion.

As a final check on the accuracy of the minimization process, a selected number of steps in a random direction away from the position of the minimum may be taken. Each time a step is taken, a new minimum is searched for. In this process, refinement of the error matrix representing the uncertainty of the final parameters is also accomplished. It has also been suggested by Cohn[32] that the final refinement of the error matrix may be facilitated by redefining the variables in the fitting problem in such a manner that the terms in the error matrix are of the same order of magnitude. This can be accomplished by modifying the VMM program to represent the a_i' and λ_i' terms as the product of new variables, q_i and r_i, and the best fit parameters previously obtained. This is equivalent to fixing the best fit parameters already obtained, defining new variables, q_i and r_i which have initial values of 1.0, and making a number of random steps in which the a_i and λ_i' of Equation 6 are represented as $q_i a_i'$ and $r_i \lambda_i'$ respectively. The final error estimates are then the product of the fixed parameters, a_i' and λ_i', and the error estimates for the parameters q_i and r_i.

The ALGOL version of Variable Metric Minimization incorporating the Integral Function Procedure, FCN, is listed on the following pages.

```
COMMENT V A R I A B L E   M E T R I C   M I N I M I Z A T I O N ;                    0000
 BEGIN       FILE OUT PUNCH 0 (2, 10);                                    100    0000
                                                          START OF SEGMENT ********** 0002
 FILE TAPE6 1 (2, 15) ; %                                                 200    0005
 FILE TAPE7 0 (2, 15) ; %                                                 300    0010
 BOOLEAN ARRAY SENSL [0 : 4] ; %                                          400    0015
 BOOLEAN ARRAY SENSW [0 : 6] ; %                                          500    0017
 REAL Q, XPR ; %                                                          600    0019
 INTEGER K ; %                                                            700    0019
 REAL ARRAY SVH [0 : 40, 0 : 40], SVX [0 : 40], SVG [0 : 40], SVS [       800    0019
 0 : 40], SVXP [0 : 40], SVGP [0 : 40], SVT [0 : 1022], SVGB [0 :         900    0025
 40], SVU [0 : 1022], SVC [0 : 10, 0 : 40], SVCTEM [0 : 40], TZERO [      1000    0033
 0 : 1022] ; %                                                            1100    0042
 REAL ARRAY SVY [0 : 1022], SVSIGMA [0 : 1022], SVYCALC [0 : 1022],       1200    0043
 SVDF [0 : 1022] ; %                                                      1300    0049
 REAL ARRAY SVA [0 : 38, 0 : 395] ; %                                     1400    0051
 INTEGER JL, JIT, JN, JM, JM1, JMS, JK, JMODE, JMM, JB;                   1500    0053
 INTEGER I, J ; %                                                         1600    0053
 REAL JGS, JE, JF, JEL, JSL, JFP, JGSP, JFB, JDELTA, JTO, JGSB, JA,       1700    0053
 JQ, JGSS, JP, JRAND, JFO, JGTT, JGTP, TAU, ELAP;                         1800    0053
 LABEL L1 ; %                                                             1900    0053
 REAL PROCEDURE FNDEXP (JX) ; %                                           2000    0053
 VALUE JX ; %                                                             2100    0053
 REAL JX ; %                                                              2200    0053
 BEGIN REAL JDEXP ; %                                                     2300    0053
                                                          START OF SEGMENT ********** 0003
   LABEL DUMMY, L1, L4, L3 ; %                                            2400    0000
   IF (ABS (JX) ≤ 0.1) THEN GO TO L1;                                     2500    0000
   JDEXP ← = 1 ; %                                                        2600    0001
   IF (JX + 25.0) > 0 THEN GO TO L4 ELSE GO TO DUMMY;                     2700    0002
   L1: JDEXP ←       (1 + JX × (.5 + JX × (.16666666667 + JX × (          2800    0004
   .41666666670-1 + JX × (.83333333330-2 + JX × (.13888888888P-2 +        2900    0006
   JX × .19841269840-3))))) × JX ;%                                       3000    0007
   GO TO DUMMY ; %                                                        3100    0012
   L4: JDEXP ← JDEXP ← EXP (JX) ; %                                       3200    0021
   L3: GO TO DUMMY ; %                                                    3300    0022
   DUMMY: FNDEXP ← JDEXP                                                  3400    0023
 END ; %                                                                  3500    0024
                                                          0003 IS 0028 LONG, NEXT SEG 0002
 REAL PROCEDURE INT (ARG1) ; %                                            3600    0053
 VALUE ARG1 ; %                                                           3700    0053
 REAL ARG1 ; %                                                            3800    0053
 BEGIN INT ← (SIGN (ARG1) × ENTIER (ABS (ARG1)))                          3900    0053
```

```
END ; %                                                          4000   0057
REAL PROCEDURE MAX (ARG1, ARG2) ; %                              4100   0062
VALUE ARG1, ARG2 ; %                                             4200   0062
REAL ARG1, ARG2 ; %                                              4300   0062
BEGIN MAX := (IF ARG1 ≥ ARG2 THEN ARG1 ELSE ARG2)                4400   0062
END ; %                                                          4500   0064
REAL PROCEDURE MIN (ARG1, ARG2) ; %                              4600   0067
VALUE ARG1, ARG2 ; %                                             4700   0067
REAL ARG1, ARG2 ; %                                              4800   0067
BEGIN MIN := (IF ARG1 ≤ ARG2 THEN ARG1 ELSE ARG2)                4900   0067
END ; %                                                          5000   0069
PROCEDURE SRMATMPY (JM, JN, SVH, SVG, SVS);                      5100   0072
INTEGER JM, JN ; %                                               5200   0072
REAL ARRAY SVH [0, 0], SVG [0], SVS [0];                         5300   0072
BEGIN INTEGER JI, JJ ; %                                         5400   0072
                                                        START OF SEGMENT ********** 0004
  LABEL DUMMY ; %                                                5500   0000
  JI := 1 ; %                                                    5600   0000
  DO                                                             5700   0000
  BEGIN SVS [JI] := 0 ; %                                        5800   0000
    JJ := 1 ; %                                                  5900   0002
    DO                                                           6000   0002
    BEGIN SVS [JI] := SVH [JI, JJ] × SVG [JJ] + SVS [JI]         6100   0002
    END UNTIL (JJ := (JJ + 1)) > JN                              6200   0005
  END UNTIL (JI := (JI + 1)) > JM ; %                            6300   0007
  GO TO DUMMY ; %                                                6400   0011
  DUMMY;                                                         6500   0011
END ; %                                                          6600   0012
                                                        0004 IS 0015 LONG, NEXT SEG 0002
PROCEDURE SRFCN (JM, SVG, JF, SVX, JM1);                         6700   0072
INTEGER JM, JM1 ; %                                              6800   0072
REAL JF ; %                                                      6900   0072
REAL ARRAY SVG [0], SVX [0] ; %                                  7000   0072
BEGIN INTEGER DX1 ; %                                            7100   0072
                                                        START OF SEGMENT ********** 0005
  INTEGER JI, JQ, JL, JJ ; %                                     7200   0000
  REAL JT1, JT2, JCHISQ, JCHI, TTAU, TEX1, XDT, DLT;             7300   0000
  REAL FX ; %                                                    7400   0000
  FORMAT FL2000 (2 I6), FL3000 (6 (F8. 1, X4)), FL4000 (/ " T=", 8   7500   0000
                                                        START OF SEGMENT ********** 0006
  E14. 6 / (/ "   ", 8 E14. 6)), FL1 (R10. 4, X4, R10. 4, X4, R15.   7600   0000
  9, X4, R10. 4), FL2 (2 I4), FL5000 (/                          7700   0000
  "          T1                                                  7800   0000
             T2             Y            SIGMA         Y CALC          "   0000
```

```
, " CHI          SUM CHISQ"), FL7000 (/ " DECAY CONSTANT", I4,                    7900    0000
" NEGATIVE, SET EQUAL TO ZERO"), FL6000 (/ " ", 7 E14. 6)$                        8000    0000
                                                                  0006 IS 0088 LONG, NEXT SEG 0005
LIST LIST1 (TZERO [I], SVU [I], SVY [I], SVSIGMA [I])$                           8100    0000
LIST LIST3 (FOR DX1 ← 1 STEP 1 UNTIL JM DO SVX [DX1])$                           8200    0011
LIST LIST4 (TZERO [JI], SVU [JI], SVY [JI], SVSIGMA [JI],                        8300    0021
SVYCALC [JI], JCHI, JCHISQ) $ %                                                  8400    0028
LIST LIST7 (JQ) $ %                                                              8500    0036
LABEL DUMMY, L8, L10, L190, L220, L250, L260, L290$                              8600    0042
LABEL L12 $ %                                                                    8700    0042
SWITCH SWGO1 ← L220, L8 $ %                                                      8800    0042
SWITCH SWGO2 ← L260, L290 $ %                                                    8900    0047
IF TIME (2) - J > 36000 THEN                                                     9000    0052
BEGIN J ← TIME (2) $ %                                                           9100    0054
  BREAK $ %                                                                      9200    0056
END $ %                                                                          9300    0057
IF (XPR ← (JM1 = 2)) > 0 THEN GO TO L250$                                        9400    0057
IF XPR = 0 THEN GO TO L10 ELSE GO TO L190$                                       9500    0059
L8: JMM ← JM DIV 2 $ %                                                           9600    0061
L10: JL ← 0 $ %                                                                  9700    0063
DO                                                                               9800    0064
BEGIN TTAU ← SVU [JL + 1] $ %                                                    9900    0064
  DLT ← TTAU - TZERO [JL + 1] $ %                                               10000    0066
  SVYCALC [JL + 1] ← 0 $ %                                                      10100    0068
  JQ ← 1 $ %                                                                    10200    0070
  DO                                                                            10300    0070
  BEGIN IF (SVX [2 × JQ] ≤ 1.0E-15) THEN                                        10400    0070
    BEGIN IF JL = 0 THEN                                                        10500    0072
      BEGIN SVX [2 × JQ] ← 0 $ %                                                10600    0073
        WRITE (TAPE6, FL7000, LIST7)                                            10700    0075
      END $ %                                                                   10800    0079
      SVYCALC [JL + 1] ← SVYCALC [JL + 1] + DLT × SVX [2 × JQ -                 10900    0079
      1] $ %                                                                    11000    0082
      SVA [2 × JQ - 1, JL + 1] ← DLT$                                           11100    0084
      SVA [2 × JQ, JL + 1] ← - TZERO [JL + 1] × DLT$                            11200    0087
      GO TO L12 $ %                                                             11300    0092
    END $ %                                                                     11400    0095
    TEX1 ← EXP (- SVX [2 × JQ] × TZERO [JL + 1])$                               11500    0095
    XDT ← SVX [2 × JQ] × DLT $ %                                                11600    0098
    FX ← FNDEXP (- XDT) $ %                                                     11700    0100
    JT1 ← - FX × TEX1 $ %                                                       11800    0102
    JT2 ← JT1 / SVX [2 × JQ] $ %                                                11900    0103
    SVYCALC [JL + 1] ← SVYCALC [JL + 1] + JT2 × SVX [2 × JQ - 1]                12000    0105
     $ %                                                                        12100    0109
```

```
    SVA (2 * JQ - 1, JL + 1) + JT2);                                    12200   0110
    SVA (2 * JQ, JL + 1) + SVX (2 * JQ - 1) * (TEX1 / SVX (2 *          12300   0113
    JQJ) * (DLT * (FX + 1) + FX * (TZERO (JL + 1) + 1 / SVX (2 *        12400   0118
    JQJ)) ; %                                                           12500   0122
    L12:                                                                12600   0124
  END UNTIL (JQ + (JQ + 1)) > JMM                                       12700   0125
END UNTIL (JL + (JL + 1)) > (JB - 1);                                   12800   0126
JF + 0 ; %                                                              12900   0130
JI + 1 ; %                                                              13000   0131
DO                                                                      13100   0131
BEGIN SVDF (JI) + SVY (JI) - SVYCALC (JI);                              13200   0131
  JF + JF + (SVDF (JI) / SVSIGMA (JI)) * 2                              13300   0134
END UNTIL (JI + (JI + 1)) > JB ; %                                      13400   0135
JJ + 1 ; %                                                              13500   0139
DO                                                                      13600   0140
BEGIN SVG (JJ) + 0 ; %                                                  13700   0140
  JI + 1 ; %                                                            13800   0141
  DO                                                                    13900   0142
  BEGIN SVG (JJ) + SVG (JJ) + (SVDF (JI) / SVSIGMA (JI) * 2) *          14000   0142
    SVA (JJ, JI)                                                        14100   0144
  END UNTIL (JI + (JI + 1)) > JB;                                       14200   0145
  SVG (JJ) + - 2 * SVG (JJ)                                             14300   0149
END UNTIL (JJ + (JJ + 1)) > JM ; %                                      14400   0150
GO TO DUMMY ; %                                                         14500   0153
L190: READ (TAPE7, FL2, JB, JMODE);                                     14600   0154
FOR I + 1 STEP 1 UNTIL JB DO READ (TAPE7, FL1, LIST1);                  14700   0167
CLOSE (TAPE7, RELEASE) ; %                                              14800   0174
GO TO SWGO1 (JMODE) ; %                                                 14900   0176
L220: JI + 2 ; %                                                        15000   0178
DO                                                                      15100   0178
BEGIN SVX (JI) + 0.69314718 / SVX (JI)                                  15200   0178
END UNTIL (JI + (JI + 2)) > JM ; %                                      15300   0179
GO TO L8 ; %                                                            15400   0183
L250: GO TO SWGO2 (JMODE) ; %                                           15500   0186
L260: JI + 2 ; %                                                        15600   0188
DO                                                                      15700   0188
BEGIN SVX (JI) + 0.69314718 / SVX (JI)                                  15800   0188
END UNTIL (JI + (JI + 2)) > JM ; %                                      15900   0189
WRITE (TAPE6, FL4000, LIST3) ; %                                        16000   0193
L290: WRITE (TAPE6, FL5000) ; %                                         16100   0196
JCHISQ + 0 ; %                                                          16200   0200
JI + 1 ; %                                                              16300   0201
DO                                                                      16400   0202
```

```
  BEGIN JCHI ← SVDF [JI] / SVSIGMA [JI];                          16500    0202
    JCHISQ ← JCHISQ + JCHI ↑ 2 ; %                                16600    0203
    WRITE (TAPE6, FL6000, LIST4)                                  16700    0205
  END UNTIL (JI ← (JI + 1)) > JB ; %                              16800    0209
  GO TO DUMMY ; %                                                 16900    0211
  DUMMY:                                                          17000    0213
END ; %                                                           17100    0213
                                                                0005 IS 0220 LONG, NEXT SEG 0002

PROCEDURE SRREADY ; %                                             17200    0072
BEGIN REAL ARRAY SVQGS [0 : 1], SVQGSP [0 : 1];                   17300    0072
                                                                START OF SEGMENT ********** 0007
  INTEGER JI ; %                                                  17400    0004
  REAL JTP1 ; %                                                   17500    0004
  COMMENT   THE FOLLOWING SUBROUTINES ARE REQUIRED:               17600    0004
          SRMATMPY, SRFCN ; %                                     17700    0004
  FORMAT FL1 (/ " UNDERSHOT") ; %                                 17800    0004
                                                                START OF SEGMENT ********** 0008
                                                                0008 IS 0006 LONG, NEXT SEG 0007
  LABEL DUMMY, L200, L201, L223, L227, L229;                      17900    0004
  SWITCH SWGD1 ← L200, L201 ; %                                   18000    0004
  GO TO SWGD1 [JL] ; %                                            18100    0009
  L200: JIT ← 1 ; %                                               18200    0012
  L201: SRMATMPY (JN, JN, SVH, SVG, SVS);                         18300    0012
  JI ← 1 ; %                                                      18400    0016
  DO                                                              18500    0017
  BEGIN SVS [JI] ← - SVS [JI]                                     18600    0017
  END UNTIL (JI ← (JI + 1)) > JN ; %                              18700    0018
  JM ← 1 ; %                                                      18800    0021
  SVQGS [1] ← 0 ; %                                               18900    0022
  JI ← 1 ; %                                                      19000    0023
  DO                                                              19100    0024
  BEGIN SVQGS [1] ← SVQGS [1] + SVS [JI] × SVG [JI]               19200    0024
  END UNTIL (JI ← (JI + 1)) > JN ; %                              19300    0025
  JGS ← SVQGS [1] ; %                                             19400    0029
  IF (JGS + JE) ≥ 0 THEN GO TO L227;                              19500    0030
  JTP1 ← - 2 × (JF / JGS) ; %                                     19600    0032
  JEL ← MIN (2, JTP1) ; %                                         19700    0034
  JSL ← - JGS ; %                                                 19800    0035
  JI ← 1 ; %                                                      19900    0036
  DO                                                              20000    0037
  BEGIN SVXP [JI] ← SVX [JI] + JEL × SVS [JI]                     20100    0037
  END UNTIL (JI ← (JI + 1)) > JN ; %                              20200    0038
  JM1 ← 2 ; %                                                     20300    0042
  SRFCN (JN, SVGP, JFP, SVXP, JM1);                               20400    0043
```

```
  JM ← 1 ; %                                                    20500   0045
  SVQGSP [1] ← 0 ; %                                            20600   0046
  JI ← 1 ; %                                                    20700   0047
  DO                                                            20800   0048
  BEGIN SVQGSP [1] ← SVQGSP [1] + SVS [JI] × SVGP [JI]          20900   0048
  END UNTIL (JI ← (JI + 1)) > JN ; %                            21000   0050
  JGSP ← SVQGSP [1] ; %                                         21100   0053
  IF JGSP ≥ 0 THEN GO TO L229 ; %                               21200   0054
  IF (JFP ≥ JF) THEN GO TO L229 ; %                             21300   0056
  WRITE (TAPE4, FL1) ; %                                        21400   0057
  JFB ← JFP ; %                                                 21500   0060
  JI ← 1 ; %                                                    21600   0061
  DO                                                            21700   0062
  BEGIN SVGB [JI] ← SVGP [JI] ; %                               21800   0062
    SVT [JI] ← SVXP [JI]                                        21900   0063
  END UNTIL (JI ← (JI + 1)) > JN ; %                            22000   0064
  IF (JEL ≥ 2) THEN GO TO L223 ; %                              22100   0067
  JL ← 3 ; %                                                    22200   0068
  GO TO DUMMY ; %                                               22300   0069
  L223: JDELTA ← 2 × JDELTA ; %                                 22400   0070
  JTU ← 1 / JSL ; %                                             22500   0071
  JL ← 2 ; %                                                    22600   0072
  GO TO DUMMY ; %                                               22700   0073
  L227: JL ← 1 ; %                                              22800   0073
  GO TO DUMMY ; %                                               22900   0074
  L229: JL ← 4 ; %                                              23000   0075
  GO TO DUMMY ; %                                               23100   0076
  DUMMY:                                                        23200   0077
END ; %                                                         23300   0078
                                                0007 IS 0082 LONG, NEXT SEG 0002

PROCEDURE SRAIM ; %                                             23400   0072
BEGIN REAL ARRAY SVQGTP [0 : 1], SVQGTT [0 : 1];                23500   0072
                                                START OF SEGMENT ********** 0009
  INTEGER JI ; %                                                23600   0004
  REAL JZ, JTI, JTPI ; %                                        23700   0004
  COMMENT   THE FOLLOWING SUBROUTINES ARE REQUIRED:             23800   0004
          SRMATMPY, SRFCN ; %                                   23900   0004
  FORMAT FL1 (/ " RICOCHET") ; %                                24000   0004
                                                START OF SEGMENT ********** 0010
                                                0010 IS 0006 LONG, NEXT SEG 0009
  LABEL DUMMY, L312, L317, L335 ; %                             24100   0004
  JZ ← JGS + JGSP + 3 × (JF - JFP) / JEL;                       24200   0004
  JTU ← JGS / JZ ; %                                            24300   0007
```

```
JTI ← JGSP / JZ ; %                                    24400   0008
JQ ← SQRT (1 - JTO × JTI) ; %                          24500   0009
JQ ← ABS (JQ × JZ) ; %                                 24600   0012
JA ← (JGSP + JQ - JZ) / (JGSP - JGS + 2 × JQ);         24700   0013
JTO ← (JEL × (JGSP + JZ + 2 × JQ) × JA * 2) / 3;       24800   0017
JFO ← JFP - JTO ; %                                    24900   0021
SRMATMPY (JN, JN, SVH, SVGP, SVT);                     25000   0022
JTP1 ← JGSP / JSL ; %                                  25100   0026
JI ← 1 ; %                                             25200   0027
DO                                                     25300   0028
BEGIN SVT [JI] ← - SVT [JI] + JTP1 × SVS [JI]          25400   0028
END UNTIL (JI ← (JI + 1)) > JN ; %                     25500   0030
JM ← 1 ; %                                             25600   0033
SVQGTP [1] ← 0 ; %                                     25700   0034
JI ← 1 ; %                                             25800   0035
DO                                                     25900   0036
BEGIN SVQGTP [1] ← SVQGTP [1] + SVT [JI] × SVGP [JI]   26000   0036
END UNTIL (JI ← (JI + 1)) > JN ; %                     26100   0038
JGTP ← SVQGTP [1] ; %                                  26200   0041
IF (2 × JTO + JGTP) < 0 THEN GO TO L317;               26300   0042
L312: JTP1 ← 1 - JA ; %                                26400   0044
JI ← 1 ; %                                             26500   0046
DO                                                     26600   0047
BEGIN SVT [JI] ← JA × SVX [JI] + JTP1 × SVXP [JI]      26700   0047
END UNTIL (JI ← (JI + 1)) > JN ; %                     26800   0049
JL ← 1 ; %                                             26900   0052
GO TO DUMMY ; %                                        27000   0053
L317: IF (JF + JGTP / 2) < 0 THEN GO TO L312;          27100   0053
JI ← 1 ; %                                             27200   0056
DO                                                     27300   0057
BEGIN SVT [JI] ← SVT [JI] + SVXP [JI]                  27400   0057
END UNTIL (JI ← (JI + 1)) > JN ; %                     27500   0058
JM1 ← 2 ; %                                            27600   0061
SRFCN (JN, SVGB, JFB, SVT, JM1);                       27700   0062
IF (JFB ≥ JFO) THEN GO TO L312 ; %                     27800   0065
WRITE (TAPE6, FL1) ; %                                 27900   0066
JI ← 1 ; %                                             28000   0069
DO                                                     28100   0070
BEGIN SVS [JI] ← SVT [JI] - SVXP [JI]                  28200   0070
END UNTIL (JI ← (JI + 1)) > JN ; %                     28300   0071
JM ← 1 ; %                                             28400   0075
SVQGTT [1] ← 0 ; %                                     28500   0075
JI ← 1 ; %                                             28600   0077
DO                                                     28700   0077
```

```
  BEGIN SVQGTT [1] + SVQGTT [1] + SVS [JI] × SVGB [JI]                    28800   0077
  END UNTIL (JI + (JI + 1)) > JN ; %                                      28900   0079
  JGTT + SVQGTT [1] ; %                                                   29000   0083
  JGIT + JGTT = JGTP ; %                                                  29100   0084
  IF JGTT < 0 THEN GO TO L335 ; %                                         29200   0085
  JGSS + JGTT ; %                                                         29300   0086
  JSL + = JGTP ; %                                                        29400   0087
  JEL + 1 ; %                                                             29500   0088
  JL + 2 ; %                                                              29600   0089
  GO TO DUMMY ; %                                                         29700   0089
  L335: JL + 3 ; %                                                        29800   0090
  GO TO DUMMY ; %                                                         29900   0091
  DUMMY:                                                                  30000   0092
END ; %                                                                   30100   0093
                                                                0009 IS 0098 LONG, NEXT SEG 0002
PROCEDURE SRFINE ; %                                                      30200   0072
BEGIN REAL ARRAY SVQGSB [0 : 1] ; %                                       30300   0072
                                                                START OF SEGMENT ********** 0011
  INTEGER JI ; %                                                          30400   0002
  REAL JTP1, JTP2 ; %                                                     30500   0002
  COMMENT   THE FOLLOWING SUBROUTINES ARE REQUIRED:                       30600   0002
          SRFCN ; %                                                       30700   0002
  FORMAT FL1 (/ " MOVE LEFT"), FL2 (/ " MOVE RIGHT")) ;                   30800   0002
                                                                START OF SEGMENT ********** 0012
                                                                0012 IS 0012 LONG, NEXT SEG 0011
  LABEL DUMMY, L413, L418, L426, L428;                                    30900   0002
  JM1 + 2 ; %                                                             31000   0002
  SRFCN (JN, SVGB, JF8, SVT, JM1)) ;                                      31100   0002
  JM + 1 ; %                                                              31200   0005
  SVQGSB [1] + 0 ; %                                                      31300   0006
  JI + 1 ; %                                                              31400   0007
  DO                                                                      31500   0008
  BEGIN SVQGSB [1] + SVQGSB [1] + SVS [JI] × SVGB [JI]                    31600   0008
  END UNTIL (JI + (JI + 1)) > JN ; %                                      31700   0010
  JGSB + SVQGSB [1] ; %                                                   31800   0013
  JTP1 + MIN (JF, JFP) ; %                                                31900   0014
  IF (JTP1 < JF8 = JE) THEN GO TO L418;                                   32000   0016
  JTP1 + JA / (1 = JA) ; %                                                32100   0017
  JTP2 + (1 = JA) / JA ; %                                                32200   0019
  JTU + JGSB × (JTP1 = JTP2) ; %                                          32300   0021
  IF (ABS (JTU) < JQ) THEN GO TO L413;                                    32400   0023
  JGSS + 2 × JQ ; %                                                       32500   0024
  JL + 1 ; %                                                              32600   0025
  GO TO DUMMY ; %                                                         32700   0026
```

```
L413: JGSS ← JTO + 2 × JQ ; %                                    32800   0027
JI ← 1 ; %                                                       32900   0028
DO                                                               33000   0029
 BEGIN SVG [JI] ← (SVGB [JI] - SVG [JI]) × JTP1 + (SVGP [JI] -   33100   0029
   SVGB [JI]) × JTP2                                             33200   0032
 END UNTIL (JI ← (JI + 1)) > JN ; %                              33300   0033
JL ← 2 ; %                                                       33400   0036
GO TO DUMMY ; %                                                  33500   0037
L418: IF (JF ≥ JFP) THEN GO TO L428;                             33600   0037
WRITE (TAPE6, FL1) ; %                                           33700   0039
JEL ← (1 - JA) × JEL ; %                                         33800   0042
JFP ← JFB ; %                                                    33900   0044
JGSP ← JGSB ; %                                                  34000   0045
JI ← 1 ; %                                                       34100   0046
DO                                                               34200   0046
 BEGIN SVXP [JI] ← SVT [JI] ; %                                  34300   0046
   SVGP [JI] ← SVGB [JI]                                         34400   0048
 END UNTIL (JI ← (JI + 1)) > JN ; %                              34500   0049
L426: JL ← 3 ; %                                                 34600   0052
GO TO DUMMY ; %                                                  34700   0052
L428: WRITE (TAPE6, FL2) ; %                                     34800   0053
JEL ← JEL × JA ; %                                               34900   0057
JF ← JFB ; %                                                     35000   0058
JGS ← JGSB ; %                                                   35100   0059
JI ← 1 ; %                                                       35200   0060
DO                                                               35300   0061
 BEGIN SVX [JI] ← SVT [JI] ; %                                   35400   0061
   SVG [JI] ← SVGB [JI]                                          35500   0062
 END UNTIL (JI ← (JI + 1)) > JN ; %                              35600   0063
 GO TO L426 ; %                                                  35700   0066
 DUMMY:                                                          35800   0066
END ; %                                                          35900   0067
                                                  0011 IS 0071 LONG, NEXT SEG 0012
PROCEDURE SRDRESS ; %                                            36000   0072
BEGIN REAL ARRAY SVQTO [0 : 1] ; %                               36100   0072
                                                  START OF SEGMENT ********** 0013
 INTEGER DX1 ; %                                                 36200   0002
 INTEGER JI, JJ ; %                                              36300   0002
 REAL JTP1 ; %                                                   36400   0002
 COMMENT   THE FOLLOWING SUBROUTINES ARE REQUIRED:               36500   0002
         SRMATMPY ; %                                            36600   0002
 FORMAT FL1 (/ " IT =", I4, "  STEP =", I4, "  F=", E14.6, "  GS="  36700   0002
                                                  START OF SEGMENT ********** 0014
```

```
, E14, 6), FL2 (/ " X=", 8 E14, 6 / (/ "   ", 8 E14, 6)), FL3 (/                36800   0002
" DELTA=", E14, 6), FL4 (/ " = = = = = = = = = ="), FL5 (/                      36900   0002
" COLINEAR"), FL6 (/ " OVERFLOW"), FL7 (/ " ERROR MATRIX"), FL8 (               37000   0002
/ " ", 8 E14, 6 / (/ " ", 8 E14, 6)), FL9 (/ " F=", E14, 6),                    37100   0002
FL10 (/ " G=", 8 E14, 6 / (/ "   ", 8 E14, 6)), FL11 (6 E12, 4));              37200   0002
                                                        0014 IS 0096 LONG, NEXT SEG 0013
LIST LIST1 (JIT, JMS, JF, JGS) ; %                                             37300   0002
LIST LIST2 (FOR DX1 ← 1 STEP 1 UNTIL JN DO SVX [DX1]);                         37400   0012
LIST LIST3 (JDELTA) ; %                                                        37500   0022
LIST LIST4 (FOR DX1 ← 1 STEP 1 UNTIL JN DO SVH [DX1, JI]);                     37600   0028
LIST LIST5 (JF) ; %                                                            37700   0039
LIST LIST6 (FOR DX1 ← 1 STEP 1 UNTIL JN DO SVG [DX1]);                         37800   0045
LIST LIST7 (FOR DX1 ← JI STEP 1 UNTIL JN DO SVH [DX1, JI]);                    37900   0055
LIST LIST8 (JDELTA, JE, JP) ; %                                                38000   0066
LABEL DUMMY, L500, L510, L519, L517, L524, L525, L543;                         38100   0075
SWITCH SWGO1 ← L500, L525, L519, L510;                                         38200   0075
GO TO SWGO1 [JL] ; %                                                           38300   0081
L500: SRMATMPY (JN, JN, SVH, SVG, SVX);                                        38400   0084
JM ← 1 ; %                                                                     38500   0087
SVQTD [1] ← 0 ; %                                                              38600   0088
JI ← 1 ; %                                                                     38700   0089
DO                                                                             38800   0090
BEGIN SVQTD [1] ← SVQTD [1] + SVX [JI] × SVG [JI]                              38900   0090
END UNTIL (JI ← (JI + 1)) > JN ; %                                             39000   0092
JTU ← SVQTD [1] ; %                                                            39100   0095
JTP1 ← JSL = JGSS * 2 / JTU = JE;                                              39200   0096
IF JTP1 < 0 THEN GO TO L524 ; %                                                39300   0099
JI ← 1 ; %                                                                     39400   0100
DO                                                                             39500   0101
BEGIN JJ ← 1 ; %                                                               39600   0101
  DO                                                                           39700   0102
  BEGIN SVH [JJ, JI] ← SVH [JJ, JI] = SVX [JI] × SVX [JJ] / JTU                39800   0102
  END UNTIL (JJ ← (JJ + 1)) > JN                                               39900   0105
END UNTIL (JI ← (JI + 1)) > JN ; %                                             40000   0108
JDELTA ← JDELTA × (JEL × JGSS / JTU);                                          40100   0111
JTU ← JEL / JGSS ; %                                                           40200   0113
L510: JI ← 1 ; %                                                               40300   0115
DO                                                                             40400   0115
BEGIN JJ ← 1 ; %                                                               40500   0115
  DO                                                                           40600   0116
  BEGIN SVH [JJ, JI] ← SVH [JJ, JI] + JTU × SVS [JI] × SVS [JJ]                40700   0116
  END UNTIL (JJ ← (JJ + 1)) > JN                                               40800   0120
END UNTIL (JI ← (JI + 1)) > JN ; %                                             40900   0122
L519: JF ← JFB ; %                                                             41000   0126
```

```
  JI ← 1 ; %                                                    41100   0126
  DO                                                            41200   0127
  BEGIN SVG [JI] ← SVGB [JI] ; %                                41300   0127
    SVX [JI] ← SVT [JI]                                         41400   0129
  END UNTIL (JI ← (JI + 1)) > JN ; %                            41500   0129
  WRITE (TAPE6, FL1, LIST1) ; %                                 41600   0132
  IF NOT SENSW [1] THEN GO TO L517;                             41700   0136
  WRITE (TAPE6, FL2, LIST2) ; %                                 41800   0137
  WRITE (TAPE6, FL3, LIST3) ; %                                 41900   0141
  L517: WRITE (TAPE6, FL4) ; %                                  42000   0145
  JIT ← JIT + 1 ; %                                             42100   0148
  JL ← 1 ; %                                                    42200   0149
  GO TO DUMMY ; %                                               42300   0150
  L524: WRITE (TAPE6, FL5) ; %                                  42400   0151
  L525: JTP1 ← JEL × JSL / JGSS ; %                             42500   0154
  JDELTA ← JDELTA × JTP1 ; %                                    42600   0156
  JTU ← (JTP1 - 1) / JSL ; %                                    42700   0158
  GO TO L510 ; %                                                42800   0159
  WRITE (TAPE6, FL6) ; %                                        42900   0160
  WRITE (TAPE6, FL7) ; %                                        43000   0163
  JI ← 1 ; %                                                    43100   0167
  DO                                                            43200   0168
  BEGIN WRITE (TAPE6, FL8, LIST4)                               43300   0168
  END UNTIL (JI ← (JI + 1)) > JN ; %                            43400   0171
  WRITE (TAPE6, FL9, LIST5) ; %                                 43500   0174
  WRITE (TAPE6, FL2, LIST2) ; %                                 43600   0177
  WRITE (TAPE6, FL10, LIST6) ; %                                43700   0181
  IF NOT SENSW [2] THEN GO TO L543;                             43800   0185
  WRITE (PUNCH, FL11, LIST2) ; %                                43900   0186
  JI ← 1 ; %                                                    44000   0190
  DO                                                            44100   0190
  BEGIN WRITE (PUNCH, FL11, LIST7)                              44200   0190
  END UNTIL (JI ← (JI + 1)) > JN ; %                            44300   0194
  WRITE (PUNCH, FL11, LIST8) ; %                                44400   0196
  L543: JL ← 2 ; %                                              44500   0200
  GO TO DUMMY ; %                                               44600   0201
  DUMMY:                                                        44700   0202
END ; %                                                         44800   0203
                                                    0013 IS 0209 LONG, NEXT SEG 0002
PROCEDURE SRRAND (JRAND) ; %                                    44900   0072
REAL JRAND ; %                                                  45000   0072
BEGIN REAL JARG2 ; %                                            45100   0072
                                                    START OF SEGMENT ********** 0015
  LABEL DUMMY ; %                                               45200   0000
```

```
  JAMG2 ← 5881 × 5881 ; %                                          45300   0000
  JRAND ← 31 × ((JRAND) MOD (JAMG2));                              45400   0001
  COMMENT  THE RANDOM NUMBER HAS SOME VALUE BETWEEN ZERO AND 5881 SQUAR  45500   0003
  ED ; %                                                           45600   0003
  COMMENT  MINUS 1.   THIS IS LEHMERS PROCEDURE, REF TODD PAGE 181;  45700   0003
  GO TO DUMMY ; %                                                  45800   0003
  DUMMY:                                                           45900   0005
END ; %                                                            46000   0005
                                                                   0015 IS 0007 LONG, NEXT SEG 0002
PROCEDURE SRSTUFF ; %                                              46100   0072
BEGIN REAL ARRAY SVQTP1 [0 : 1] ; %                                46200   0072
                                                                   START OF SEGMENT ********** 0016
  INTEGEN JI ; %                                                   46300   0002
  REAL JDIV, JTP1 ; %                                              46400   0002
  COMMENT   THE FOLLOWING SUBROUTINES ARE REQUIRED:                46500   0002
        SRRAND, SRMATMPY ; %                                       46600   0002
  FORMAT FL1 (/ " RANDOM STEP ", I4, "  GS=", E14. 6);            46700   0002
                                                                   START OF SEGMENT ********** 0017
                                                                   0017 IS 0010 LONG, NEXT SEG 0016
  LIST LIST1 (JMS, JGS) ; %                                        46800   0002
  LABEL DUMMY, L617 ; %                                            46900   0010
  JK ← JK - 1 ; %                                                  47000   0010
  IF JK < 0 THEN GO TO L617 ; %                                    47100   0011
  JMS ← JMS + 1 ; %                                                47200   0012
  WRITE (TAPE6, FL1, LIST1) ; %                                    47300   0013
  JRAND ← 1 ; %                                                    47400   0017
  JI ← 1 ; %                                                       47500   0018
  DO                                                               47600   0019
  BEGIN SRRAND (JRAND) ; %                                         47700   0019
    JDIV ← (5881 × 5881) - 1 ; %                                   47800   0019
    SVT [JI] ← (JRAND / JDIV) - 0,5                                47900   0021
  END UNTIL (JI ← (JI + 1)) > JN ; %                               48000   0022
  SRMATMPY (JN, JN, SVH, SVT, SVS);                                48100   0026
  JN ← 1 ; %                                                       48200   0029
  SVQTP1 [1] ← 0 ; %                                               48300   0030
  JI ← 1 ; %                                                       48400   0031
  DO                                                               48500   0032
  BEGIN SVQTP1 [1] ← SVQTP1 [1] + SVS [JI] × SVT [JI]              48600   0032
  END UNTIL (JI ← (JI + 1)) > JN ; %                               48700   0034
  JTP1 ← SVQTP1 [1] ; %                                            48800   0037
  JTP1 ← SQRT (JTP1) ; %                                           48900   0038
  JP ← SQRT (JP / 5) ; %                                           49000   0039
  JEL ← JP / JTP1 ; %                                              49100   0041
  JI ← 1 ; %                                                       49200   0042
  DO                                                               49300   0043
```

```
  BEGIN SVX [JI] ← SVX [JI] + JEL × SVS [JI]                    49400   0043
  END UNTIL (JI ← (JI + 1)) > JN ; %                            49500   0045
  JM1 ← 2 ; %                                                   49600   0048
  JL ← 1 ; %                                                    49700   0049
  GO TO DUMMY ; %                                               49800   0050
  L617: JL ← 2 ; %                                              49900   0053
  JMS ← 0 ; %                                                   50000   0053
  GO TO DUMMY ; %                                               50100   0054
  DUMMY:                                                        50200   0055
END ; %                                                         50300   0055
                                                    0016 IS 0059 LONG, NEXT SEG 0002

PROCEDURE MAINPRO ; %                                           50400   0072
BEGIN INTEGER DX1 ; %                                           50500   0072
                                                    START OF SEGMENT ********** 0018
  INTEGER JNC, JI, JJ, JJ1 ; %                                  50600   0000
  REAL JFAC, TO, T1, PROC, INOUT ; %                            50700   0000
  COMMENT   THE FOLLOWING SUBROUTINES ARE REQUIRED:             50800   0000
          SRFCN, SRREADY, SRAIM, SRFIRE, SRDRESS, SRSTUFF;      50900   0000
  FORMAT FL1 (                                                  51000   0000
                                                    START OF SEGMENT ********** 0019
  " IDENTIFICATION TITLE                                "               0000
  , "          "), FL2 (3 I6, F6. 1), FL3 (6 E12. 4), FL5 (     51200   0000
  " VARIABLE METRIC MINIMIZATION"), FL6 (/ " N=", I2, "  K=", I2,  51300   0000
  "  E=", E14. 6, "  P=", E14. 6, "  DELTA=", E14. 6), FL7 (/       51400   0000
  " X=", 8 E14. 6 / (/ "   ", 8 E14. 6)), FL8 (/ " H"), FL9 (/ " "  51500   0000
  , 8 E14. 6 / (/ " ", 8 E14. 6)), FL14 (/ " IT ", I4, "  STEP ",  51600   0000
  I4, "  F=", E14. 6), FL4 (/ " - - - - - - - - - -"), FL10 (/     51700   0000
  " FINAL VALUES"), FL11 (/ " ERROR MATRIX"), FL20 (/               51800   0000
  "PROCESSOR TIME = ", F6. 2, "SECONDS      I/O TIME = ", F6. 2),   51900   0000
  FL13 (/ " F=", E14. 6, "  GS=", E14. 6), FL12 (/ " G=", 8 E14. 6  52000   0000
  / (/ "   ", 8 E14. 6)) ; %                                      52100   0000
                                                    0019 IS 0146 LONG, NEXT SEG 0018
  LIST LIST1 (JN, JK, JNC, JFAC) ; %                            52200   0000
  LIST LIST2 (FOR DX1 ← 1 STEP 1 UNTIL JN DO SVX [DX1]);        52300   0010
  LIST LIST3 (FOR DX1 ← JI STEP 1 UNTIL JN DO SVH [DX1, JI]);   52400   0020
  LIST LIST4 (JDELTA, JE, JP) ; %                               52500   0031
  LIST LIST5 (FOR DX1 ← 1 STEP 1 UNTIL JN DO SVC [JJ, DX1]);    52600   0040
  LIST LIST6 (JN, JK, JE, JP, JDELTA);                          52700   0051
  LIST LIST7 (FOR DX1 ← 1 STEP 1 UNTIL JN DO SVH [DX1, JI]);    52800   0062
  LIST LIST8 (JIT, JMS, JF) ; %                                 52900   0073
  LIST LIST9 (JF, JGS) ; %                                      53000   0082
  LIST LIST10 (FOR DX1 ← 1 STEP 1 UNTIL JN DO SVG [DX1]);       53100   0090
  LIST LIST20 (PROC, INOUT) ; %                                 53200   0100
  LABEL L101, L1064, L1065, L109, L1165, L1201, L1209, L121, L124,  53300   0108
```

```
L126, L129, L132, L133, L135, L137, L159, L139, L1393, L1395,          53400   0108
L142, L157, L162, LEND ; %                                             53500   0108
SWITCH SWGO1 ← L139, L159, L137, L126;                                 53600   0108
SWITCH SWGO2 ← L129, L135, L137;                                       53700   0114
SWITCH SWGO3 ← L135, L132, L126;                                       53800   0119
SWITCH SWGO4 ← L124, L162 ; %                                          53900   0124
SWITCH SWGO5 ← L1165, L142 ; %                                         54000   0129
L101: JMS ← 0 ; %                                                      54100   0134
T1 ← TIME (3) ; %                                                      54200   0135
T0 ← TIME (2) ; %                                                      54300   0137
READ (TAPE7, FL1) [L1] ; %                                             54400   0138
READ (TAPE7, FL2, LIST1) [L1] ; %                                      54500   0143
READ (TAPE7, FL3, LIST2) [L1] ; %                                      54600   0148
JI ← 1 ; %                                                             54700   0153
DO                                                                     54800   0153
BEGIN JJ ← JI ; %                                                      54900   0153
  DO                                                                   55000   0154
  BEGIN SVH [JJ, JI] ← 0                                               55100   0154
  END UNTIL (JJ ← (JJ + 1)) > JN                                       55200   0155
END UNTIL (JI ← (JI + 1)) > JN ; %                                     55300   0157
JI ← 1 ; %                                                             55400   0161
DO                                                                     55500   0161
BEGIN IF JFAC ≤ 0 THEN GO TO L1064;                                    55600   0161
  SVH [JI, JI] ← JFAC ; %                                              55700   0163
  GO TO L1065 ; %                                                      55800   0165
  L1064: READ (TAPE7, FL3, LIST3);                                     55900   0169
  L1065:                                                               56000   0173
END UNTIL (JI ← (JI + 1)) > JN ; %                                     56100   0173
JI ← 1 ; %                                                             56200   0175
DO                                                                     56300   0176
BEGIN JJ ← JI ; %                                                      56400   0176
  DO                                                                   56500   0176
  BEGIN SVH [JI, JJ] ← SVH [JJ, JI]                                    56600   0176
  END UNTIL (JJ ← (JJ + 1)) > JN                                       56700   0179
END UNTIL (JI ← (JI + 1)) > JN ; %                                     56800   0181
READ (TAPE7, FL3, LIST4) ; %                                           56900   0184
IF JNC ≤ 0 THEN GO TO L109 ; %                                         57000   0188
JJ ← 1 ; %                                                             57100   0189
DO                                                                     57200   0190
BEGIN READ (TAPE7, FL3, LIST5)                                         57300   0190
END UNTIL (JJ ← (JJ + 1)) > JNC;                                       57400   0193
L109: WRITE (TAPE6, FL5) ; %                                           57500   0196
WRITE (TAPE6, FL1) ; %                                                 57600   0200
WRITE (TAPE6, FL6, LIST6) ; %                                          57700   0204
```

```
WRITE (TAPE6, FL7, LIST2) ; %                                    57800   0207
WRITE (TAPE6, FL8) ; %                                           57900   0211
JI ← 1 ; %                                                       58000   0215
DO                                                               58100   0215
BEGIN WRITE (TAPE6, FL9, LIST7)                                  58200   0215
END UNTIL (JI ← (JI + 1)) > JN ; %                               58300   0219
JM1 ← 1 ; %                                                      58400   0221
L1165: JF ← 0 ; %                                                58500   0222
SRFCN (JN, SVG, JF, SVX, JM1) ; %                                58600   0223
IF JF ≤ 0 THEN GO TO L1393 ; %                                   58700   0226
JL ← 1 ; %                                                       58800   0227
JIT ← 0 ; %                                                      58900   0228
WRITE (TAPE6, FL14, LIST8) ; %                                   59000   0229
WRITE (TAPE6, FL7, LIST2) ; %                                    59100   0233
WRITE (TAPE6, FL4) ; %                                           59200   0236
L1201: IF JNC ≤ 0 THEN GO TO L121;                               59300   0240
JJ1 ← 1 ; %                                                      59400   0242
DO                                                               59500   0243
BEGIN JI ← 1 ; %                                                 59600   0243
  DO                                                             59700   0243
  BEGIN SVCTEM [JI] ← SVC [JJ1, JI];                             59800   0243
    SVT [JI] ← 0                                                 59900   0246
  END UNTIL (JI ← (JI + 1)) > JN;                                60000   0246
  JI ← 1 ; %                                                     60100   0249
  DO                                                             60200   0250
  BEGIN JJ ← 1 ; %                                               60300   0250
    DO                                                           60400   0251
    BEGIN SVT [JI] ← SVT [JI] + SVH [JJ, JI] × SVCTEM [JJ]       60500   0251
    END UNTIL (JJ ← (JJ + 1)) > JN                               60600   0253
  END UNTIL (JI ← (JI + 1)) > JN;                                60700   0256
  JTO ← 0 ; %                                                    60800   0259
  JI ← 1 ; %                                                     60900   0260
  DO                                                             61000   0260
  BEGIN JTO ← JTO + SVCTEM [JI] × SVT [JI]                       61100   0260
  END UNTIL (JI ← (JI + 1)) > JN;                                61200   0261
  IF (JTO ≤ JE) THEN GO TO L1209;                                61300   0265
  JI ← 1 ; %                                                     61400   0266
  DO                                                             61500   0267
  BEGIN JJ ← 1 ; %                                               61600   0267
    DO                                                           61700   0268
    BEGIN SVH [JJ, JI] ← SVH [JJ, JI] - SVT [JI] × SVT [JJ] /    61800   0268
      JTO                                                        61900   0271
    END UNTIL (JJ ← (JJ + 1)) > JN                               62000   0271
  END UNTIL (JI ← (JI + 1)) > JN;                                62100   0274
  L1209:                                                         62200   0277
```

```
END UNTIL (JJ1 + (JJ1 + 1)) > JNC);                    62300   0281
L121: SRREADY ; %                                      62400   0283
GO TO SWGO1 [JL] ; %                                   62500   0284
L124: JL + 2 ; %                                       62600   0286
GO TO L1201 ; %                                        62700   0287
L126: SRAIM ; %                                        62800   0288
GO TO SWGO2 [JL] ; %                                   62900   0289
L129: SRFIRE ; %                                       63000   0291
GO TO SWGO3 [JL] ; %                                   63100   0292
L132: JL + 1 ; %                                       63200   0294
L133: SRDRESS ; %                                      63300   0295
GO TO SWGO4 [JL] ; %                                   63400   0296
L135: JL + 2 ; %                                       63500   0298
GO TO L133 ; %                                         63600   0299
L137: JL + 3 ; %                                       63700   0300
GO TO L133 ; %                                         63800   0301
L159: JL + 4 ; %                                       63900   0302
GO TO L133 ; %                                         64000   0303
L139: IF SENSW [1] THEN GO TO L1395;                   64100   0304
L1393: WRITE (TAPE6, FL7, LIST2));                     64200   0306
L1395: SRSTUFF ; %                                     64300   0310
GO TO SWGO5 [JL] ; %                                   64400   0311
L142: WRITE (TAPE6, FL10) ; %                          64500   0313
WRITE (TAPE6, FL11) ; %                                64600   0317
JI + 1 ; %                                             64700   0321
DO                                                     64800   0321
BEGIN WRITE (TAPE6, FL9, LIST7)                        64900   0321
END UNTIL (JI + (JI + 1)) > JN ; %                     65000   0325
WRITE (TAPE6, FL13, LIST9) ; %                         65100   0327
WRITE (TAPE6, FL7, LIST2) ; %                          65200   0331
WRITE (TAPE6, FL12, LIST10) ; %                        65300   0335
JM1 + 3 ; %                                            65400   0339
SRFCN (JN, SVG, JF, SVX, JM1) ; %                      65500   0339
PROC + (TIME (2) - T0) / 60 ; %                        65600   0342
INOUT + (TIME (3) - T1) / 60 ; %                       65700   0344
IF NOT SENSW [2] THEN GO TO L157;                      65800   0347
WRITE (PUNCH, FL3, LIST2) ; %                          65900   0348
JI + 1 ; %                                             66000   0351
DO                                                     66100   0352
BEGIN WRITE (PUNCH, FL3, LIST3)                        66200   0352
END UNTIL (JI + (JI + 1)) > JN ; %                     66300   0356
WRITE (PUNCH, FL3, LIST4) ; %                          66400   0358
L157: WRITE (TAPE6, FL20, LIST20));                    66500   0362
GO TO LEND ; %                                         66600   0366
```

```
  L102: WRITE (TAPE6, FL20, LIST20);                    66700   0367
  GO TO LEND; %                                         66800   0371
  LEND:                                                 66900   0372
END; %                                                  67000   0373
                                                 0018 IS 0380 LONG, NEXT SEG 0002
COMMENT  INITIALIZING BLOCK; %                          67100   0072
XPR + Q + K + O; %                                      67200   0072
SENSW [1] + TRUE; %                                     67300   0073
SENSW [2] + TRUE; %                                     67400   0075
SENSW [3] + FALSE; %                                    67500   0076
SENSW [4] + FALSE; %                                    67600   0077
SENSW [5] + FALSE; %                                    67700   0078
SENSW [6] + FALSE; %                                    67800   0080
SENSL [1] + FALSE; %                                    67900   0081
SENSL [2] + FALSE; %                                    68000   0082
SENSL [3] + FALSE; %                                    68100   0083
SENSL [4] + FALSE; %                                    68200   0085
MAINPRO; %                                              68300   0086
L1: END.                                                68400   0086
```

APPENDIX F

PROCEDURE RSLTS

Following the analysis by least squares fitting for the parameters which best describe the flux decay data obtained after a Georgia Tech Research Reactor rod drop, it is desirable to adjust the resulting parameters so that they may be independent of the power history before the experiment and the details of shutdown. This is also necessary if one desires to compare them with small-sample data reported in the literature. This adjustment is necessary due to:

1. non-saturation of delayed neutron precursors
2. post-rod-drop sub-critical multiplication of neutrons resulting in additional fissions after shutdown
3. effective zero-time correction.

The non-saturation of precursors results in a simple absence of delayed neutron producers, hence a decrease in observed abundance. Additional fissions after shutdown produce analyzed periods which are longer than the true periods and result in changed relative group abundances as well. The zero-time correction was discussed in Appendix C. The correction terms which are applied in the B-5500 ALGOL Procedure RSLTS will now be outlined.

In the reactor which has been operated at some equilibrium level, n_o, without external source and with $k_o = 1$, the reactor kinetics equations become:

$$\frac{dn}{dt} = \frac{k_{p_o} - 1}{\ell^*} n_o + \sum_i \lambda_i C_{i_o} = 0 \qquad (7)$$

$$\frac{dC_i}{dt} = \sum_i \frac{\beta_i^{eff} n_o}{\ell^*} - \lambda_i C_{i_o} = 0$$

where k_{p_o} is the prompt reproduction number.

From Equation 7,

$$n_o = \frac{\ell^* \sum_i \lambda_i C_{i_o}}{1 - k_{p_o}} = \frac{\ell^* \sum_i \lambda_i C_{i_o}}{\sum_i \beta_i^{eff}}$$

Following a negative reactivity insertion, $-\Delta k$, such as occurs in a rod drop, there is a drop within several prompt neutron lifetimes to a lower neutron level determined by the new prompt reproduction number, k_{p_1}. Keepin[23] refers to this lower neutron level as "quasistatic" since it is nearly constant on a time scale encompassing many prompt neutron lifetimes. The level finally begins to be decreased by delayed neutron precursor decay, but just after the prompt drop the instantaneous value of dn/dt is approximately zero and the time zero precursor concentrations, C_{i_o}, are essentially unchanged. Thus

$$\frac{dn}{dt} \cong \frac{k_{p_1} - 1}{\ell^*} n_1 + \sum_i \lambda_i C_{i_o} = 0$$

$$n_1 \cong \frac{\ell^* \sum_i \lambda_i C_{i_o}}{1 - k_{p_o}}$$

$$\frac{n_1}{n_o} \cong \frac{1 - k_{p_o}}{1 - k_{p_1}} \cong \frac{\sum_i \beta_i \text{eff}}{\sum_i \beta_i \text{eff} + \Delta k} \tag{8}$$

Since $$n(t)/n_o = \sum_i a_i e^{-\lambda_i' t} \tag{9}$$

describes the decay in neutron level following a rod drop, with the zero-time for Equation 9 coinciding with time 0 (i.e. the time before the prompt drop) in Equation 8, then

$$\sum_i a_i = \frac{\sum_i \beta_i \text{eff}}{\sum_i \beta_i \text{eff} + \Delta k}$$

The summation is on i over the entire range of delayed neutron precursors. Therefore,

$$a_i = \frac{\beta_i \text{eff}}{\sum_i \beta_i \text{eff} + \Delta k} \tag{10}$$

assuming that each delayed neutron precursor group is at equilibrium and the zero-time is precise. In order to account for non-equilibrium, Equa-

tion 10 may be modified:

$$a_i = \frac{\beta_i \text{eff}(1 - e^{-\lambda_i t})}{\sum_i \beta_i \text{eff} + \Delta k} \tag{11}$$

where t is the time of operation at a steady power before the rod drop. This formulation assumes that the equilibrium fraction for the i^{th} precursor is adequately determined by the $(1 - e^{-\lambda_i t})$ buildup term, which in fact ignores power history other than that at constant power immediately preceding the rod drop. In the execution of Procedure RSLTS, the equilibrium fraction is accounted for as described in Appendix B.

The application of RP-129 and RODROP permitted an assessment of the constructive zero-time of an instantaneous reactivity insertion equivalent to the actual insertion. These techniques are discussed in Appendix C. The difference between constructive zero-time and the zero-time used for curve fitting (the time the control blade hit its down-limit switch) should be applied to the abundance of each group for enhancement of the abundance value. This enhancement factor is $e^{\lambda_i(t_d - t_o)}$, where t_d is the assumed drop time and t_o is the constructive zero-time. Thus, the final value of effective delayed neutron relative abundance may be expressed as

$$R_i = \frac{a_i' e^{\lambda_i(t_d - t_o)}/Q_i}{\sum_i a_i' e^{\lambda_i(t_d - t_o)}/Q_i} \tag{12}$$

The analyzed decay constants, λ_i', which appear in Equation 9 are smaller than the λ_i in Equation 11 since the power level decays more slowly due to precursors generated after the rod drop. The fractional decrease in λ_i is directly related to the i^{th} group fraction of the normalized power level after the prompt drop.

$$\frac{\lambda_i - \lambda_i'}{\lambda_i} = \frac{\beta_i eff}{\sum_i \beta_i eff + \Delta k}$$

Then
$$\lambda_i = \lambda_i'/\left(1 - \frac{\beta_i eff}{\sum_i \beta_i eff + \Delta k}\right) \qquad (13)$$

Since
$$\frac{\beta_i eff}{\sum_i \beta_i eff + \Delta k} = \frac{a_i' e^{\lambda_i (t_d - t_o)}}{Q_i}$$

Equation 13 would not appear capable of explicit solution for λ_i. An iterative procedure might have been employed in which the first λ_i used to determine the value of Q_i required in Equation 12 was λ_i'. This is a good first guess since a single rod drop of $\Delta k = 0.075$ produces a maximum value of

$$\frac{\beta_i eff}{\sum_i \beta_i eff + \Delta k} \cong \frac{0.0035}{0.0075 + 0.075} \cong 0.043$$

and generally much smaller values. In practice, all four shim rods were

dropped and the value of $\frac{\beta_i eff}{\sum_i \beta_i eff + \Delta k}$ was sufficiently small that the equilibrium fraction value could be determined by a single pass through Procedure BLDUP using λ_i' .

With the value of $\frac{\beta_i eff}{\sum_i \beta_i eff + \Delta k}$ thus determined, the final value of λ_i was obtained directly from Equation 13.

```
COMMENT   R S L T S ;                                                    0000
 BEGIN FILE OUT TAPE6 1 (2, 15) ; %                                 100  0000
                                                  START OF SEGMENT ********** 0002
 FILE IN TAPE7 0 (2, 15) ; %                                        200  0005
 FILE OUT PUNCH 0 (2, 10) ; %                                       300  0010
 INTEGER ARRAY TYPE [0 : 1022], T [0 : 1022];                       400  0015
 REAL ARRAY SVH [0 : 40, 0 : 40], SVX [0 : 40], SVXP [0 : 40],      500  0019
 EQFRAC [0 : 30], RELAB [0 : 30], BIEM [0 : 30], CONCRI [0 : 40],   600  0025
 PHI [0 : 1022], ENH [0 : 40], ERROR [0 : 40];                      700  0033
 INTEGER I, J, C, NP, JI, IND, TD, NOINT, D, K;                     800  0039
 INTEGER CONIND, TPREV, TCON, Z, M, TSTART;                         900  0039
 REAL BSUM, PHID, DTSTART, NZERO ; %                               1000  0039
 PROCEDURE BLOUP (IND, I, LAM, EQFRAC);                            1100  0039
 INTEGER IND, I ; %                                                1200  0039
 REAL ARRAY LAM [0], EQFRAC [0] ; %                                1300  0039
 BEGIN REAL SUMINT, FAC1, FAC2, FAC3, FAC5, FAC6, FAC7, L, E;      1400  0039
                                                  START OF SEGMENT ********** 0003
   REAL ARRAY CZERO [0 : 30], CR [0 : 30], THA [0 : 30], TTMIN [0 : 1500  0000
   30], CI [0 : 30] ; %                                            1600  0006
   LABEL L1, L8, L2, L3, L4, LA, L5, L6, L7;                       1700  0010
   FORMAT FL10 (I10), FL11 (I1, I6, R17. 2, I1, I6, R17. 2, I1, I6, 1800  0010
                                                  START OF SEGMENT ********** 0004
   R17. 2), FL12 (I10, R12. 4), FL13 (I10, X10, E12. 4);           1900  0010
                                                  0004 IS 0027 LONG, NEXT SEG 0003
   LIST LIST10 (NOINT) ; %                                         2000  0010
   LIST LIST11 (FOR K = 1 STEP 1 UNTIL NOINT DO [TYPE [K], T [K],  2100  0016
   PHI [K]]) ; %                                                   2200  0022
   LIST LIST12 (TD, PHID) ; %                                      2300  0029
   LIST LIST13 (I, EQFRAC [I]) ; %                                 2400  0037
   TCON = 0.0 ; %                                                  2500  0045
   TPREV = 0.0 ; %                                                 2600  0045
   IF IND = 0 THEN GO TO L1 ; %                                    2700  0046
   FOR K = 1 STEP 1 UNTIL NP DO CZERO [K] = 0.0;                   2800  0047
   READ (TAPE7, FL10, LIST10) ; %                                  2900  0052
   READ (TAPE7, FL11, LIST11) ; %                                  3000  0056
   READ (TAPE7, FL12, LIST12) ; %                                  3100  0060
   FOR K = 1 STEP 1 UNTIL NOINT DO T [K] = T [K] * 60;             3200  0064
   TD = TD * 60 ; %                                                3300  0073
   L1: T [NOINT + 1] = TD ; %                                      3400  0074
   PHI [NOINT + 1] = PHID ; %                                      3500  0076
   CR [I] = (PHID / LAM [I]) ; %                                   3600  0078
   THA [I] = 0.693 / LAM [I] ; %                                   3700  0080
   TTMIN [I] = TD - 6 * THA [I] ; %                                3800  0082
   SUMINT = 0 ; %                                                  3900  0085
   Z = NOINT ; %                                                   4000  0085
```

```
   IF (TTMIN [I] ≤ TCON) THEN GO TO L4;                                    4100    0086
   L2: IF (TTMIN [I] ≥ T [Z]) THEN GO TO L3;                               4200    0088
   Z ← Z - 1 ; %                                                           4300    0089
   GO TO L2 ; %                                                            4400    0091
   L3: TSTART ← T [Z] ; %                                                  4500    0093
   GO TO LA ; %                                                            4600    0094
   L4: TSTART ← TCON ; %                                                   4700    0094
   Z ← 1 ; %                                                               4800    0095
   LA: FOR M ← Z STEP 1 UNTIL NOINT DO                                     4900    0096
   BEGIN IF (TYPE [M] ≠ 0) THEN GO TO L6;                                  5000    0098
     L5: FAC1 ← T [M + 1] - T [M] ; %                                      5100    0099
     FAC2 ← EXP (LAM [I] × (T [M + 1] - TO));                              5200    0102
     FAC3 ← EXP (LAM [I] × (T [M] - TO));                                  5300    0105
     FAC5 ← (T [M + 1] × PHI [M] - T [M] × PHI [M + 1]) / FAC1;             5400    0108
     FAC6 ← FAC2 - FAC3 ; %                                                5500    0113
     FAC7 ← (PHI [M + 1] - PHI [M]) / FAC1;                                5600    0114
     L ← FAC5 × FAC6 / LAM [I] + FAC7 × ((T [M + 1] × FAC2 - T [M]         5700    0117
     × FAC3) / LAM [I] - FAC6 / LAM [I] × LAM [I]);                        5800    0120
     SUMINT ← SUMINT + L ; %                                               5900    0125
     GO TO L7 ; %                                                          6000    0126
     L6: FAC1 ← T [M + 1] - T [M] ; %                                      6100    0127
     FAC2 ← EXP (LAM [I] × (T [M + 1] - TO));                              6200    0129
     FAC3 ← EXP (LAM [I] × (T [M] - TO));                                  6300    0132
     FAC6 ← LN (PHI [M + 1] / PHI [M]);                                    6400    0135
     E ← FAC1 × PHI [M] × (1 / (FAC6 ← LAM [I] × FAC1)) × ((PHI [M         6500    0138
     + 1] / PHI [M]) × FAC2 - FAC3);                                       6600    0141
     SUMINT ← SUMINT + E ; %                                               6700    0144
     L7:                                                                   6800    0146
   END ; %                                                                 6900    0146
   CT [I] ← CZERO [I] × EXP (- LAM [I] × (TSTART - TPREV)) + SUMINT        7000    0148
    ; %                                                                    7100    0151
   EQFRAC [I] ← CT [I] / CR [I] ; %                                        7200    0152
   L8: WRITE (TAPE6, FL13, LIST13);                                        7300    0155
 END ; %                                                                   7400    0158
                                                    0003 IS 0167 LONG, NEXT SEG 0002
PROCEDURE RSLTS;                                                         7500    0039
 BEGIN LABEL L1, L4, LFIN ; %                                            7600    0039
                                                    START OF SEGMENT ********** 0005
FORMAT ID1(                                                                      0000
                                                    START OF SEGMENT ********** 0006
   " IDENTIFICATION TITLE                                          ",            0000
   "          "));                                                              0000
                                                    0006 IS 0016 LONG, NEXT SEG 0005
```

```
   FORMAT FL1 (I2), FL2 (6 R12. 4), FL3 (/,                                7700    0000
                                                  START OF SEGMENT ********** 0007
   "GROUP   APRIME    ENHANCEMENT    REL ABUND   ERROR ESTIMATE     OR    7800    0000
IGINAL LAMBDA    CORRECTED LAMBDA    ERROR ESTIMATE"                             0000
   ), FL4 (/ /, X2, I2, X1, R11. 4, X1, R12. 5, X2, E12. 5, X2,            7900    0000
   E12. 5, X7, E11. 4, X8, E11. 4, X7, E11. 4)$                            8000    0000
                                                  0007 IS 0053 LONG, NEXT SEG 0005
   LIST LIST2 (FOR I ← 1 STEP 1 UNTIL NP DO SVX [I])$                      8100    0000
   LIST LIST3 (FOR I ← JI STEP 1 UNTIL NP DO SVH [I, JI])$                 8200    0010
   LIST LIST4 (K, SVXP [I], ENH [I], RELAB [I], ERROR [I], SVXP [J]        8300    0021
   , SVX [J], ERROR [J]) $ %                                               8400    0030
   C ← 0 $ %                                                               8500    0038
   IND ← 1 $ %                                                             8600    0038
   BSUM ← 0 $ %                                                            8700    0039
READ(TAPE7,ID1)$                                                                   0040
WRITE(TAPE6,ID1)$                                                                  0044
   READ (TAPE7, FL1, NP) $ %                                               8800    0047
   READ (TAPE7, FL2, DTSTART, NZERO)$                                      8900    0058
   READ (TAPE7, FL2, LIST2) $ %                                            9000    0070
   WRITE (TAPE6, FL2, LIST2) $ %                                           9100    0074
   FOR JI ← 1 STEP 1 UNTIL NP DO READ (TAPE7, FL2, LIST3)$                 9200    0078
   L1: C ← C + 1 $ %                                                       9300    0086
   FOR I ← 1 STEP 2 UNTIL (NP - 1) DO                                      9400    0088
   BEGIN SVXP [I] ← SVX [I] $ %                                            9500    0092
     SVXP [I + 1] ← SVX [I + 1]                                            9600    0094
   END $ %                                                                 9700    0095
   FOR I ← 1 STEP 2 UNTIL (NP - 1) DO                                      9800    0098
   BEGIN L4: C ← C + 1 $ %                                                 9900    0102
     D ← I + 1 $ %                                                        10000    0104
     BLOUP (IND, D, SVXP, EQFRAC) $ %                                     10100    0105
     IND ← D $ %                                                          10200    0108
     ENH [I] ← EXP (SVX [I + 1] × DTSTART) / EQFRAC [I + 1]$              10300    0108
     BTEM [I] ← SVXP [I] × ENH [I]$                                       10400    0113
     SVX [I] ← BTEM [I] $ %                                               10500    0115
     LFIN: C ← 0 $ %                                                      10600    0116
     BSUM ← BSUM + SVX [I]                                                10700    0117
   END $ %                                                                10800    0118
   FOR I ← 1 STEP 2 UNTIL (NP - 1) DO                                     10900    0119
   BEGIN RELAB [I] ← SVX [I] / BSUM$                                      11000    0124
     ERROR [I] ← SQRT (SVH [I, I]) × ENH [I] / BSUM$                      11100    0126
     ERROR [I + 1] ← SQRT (SVH [I + 1, I + 1])$                           11200    0130
     SVX [I + 1] ← SVXP [I + 1] / (1 - BTEM [I] / NZERO)                  11300    0134
   END $ %                                                                11400    0137
```

```
    WRITE (TAPE6, FL3) ; %                                    11500    0139
    FOR K ← 1 STEP 1 UNTIL (NP / 2) DO                        11600    0142
    BEGIN J ← 2 × K ; %                                       11700    0146
      I ← J - 1 ; %                                           11800    0148
      WRITE (TAPE6, FL4, LIST4)                               11900    0149
    END ; %                                                   12000    0152
  END ; %                                                     12100    0153
                                                  0005 IS 0156 LONG, NEXT SEG 0002
RSLTS    END. %                                             12200    0039
```

APPENDIX G

SCHEMATIC DIAGRAMS OF ELECTRONIC UNITS

The following pages in this appendix are devoted to schematic diagrams which describe the electronic circuits developed in the course of this work. These designs, while they have been influenced and in some cases specified by the author of this thesis, are not his own. They are the designs of Mr. J. C. Gundlach of Reactor Controls, Inc. and Messrs. R. E. Meek and T. L. Erb of the Georgia Tech Nuclear Research Center.

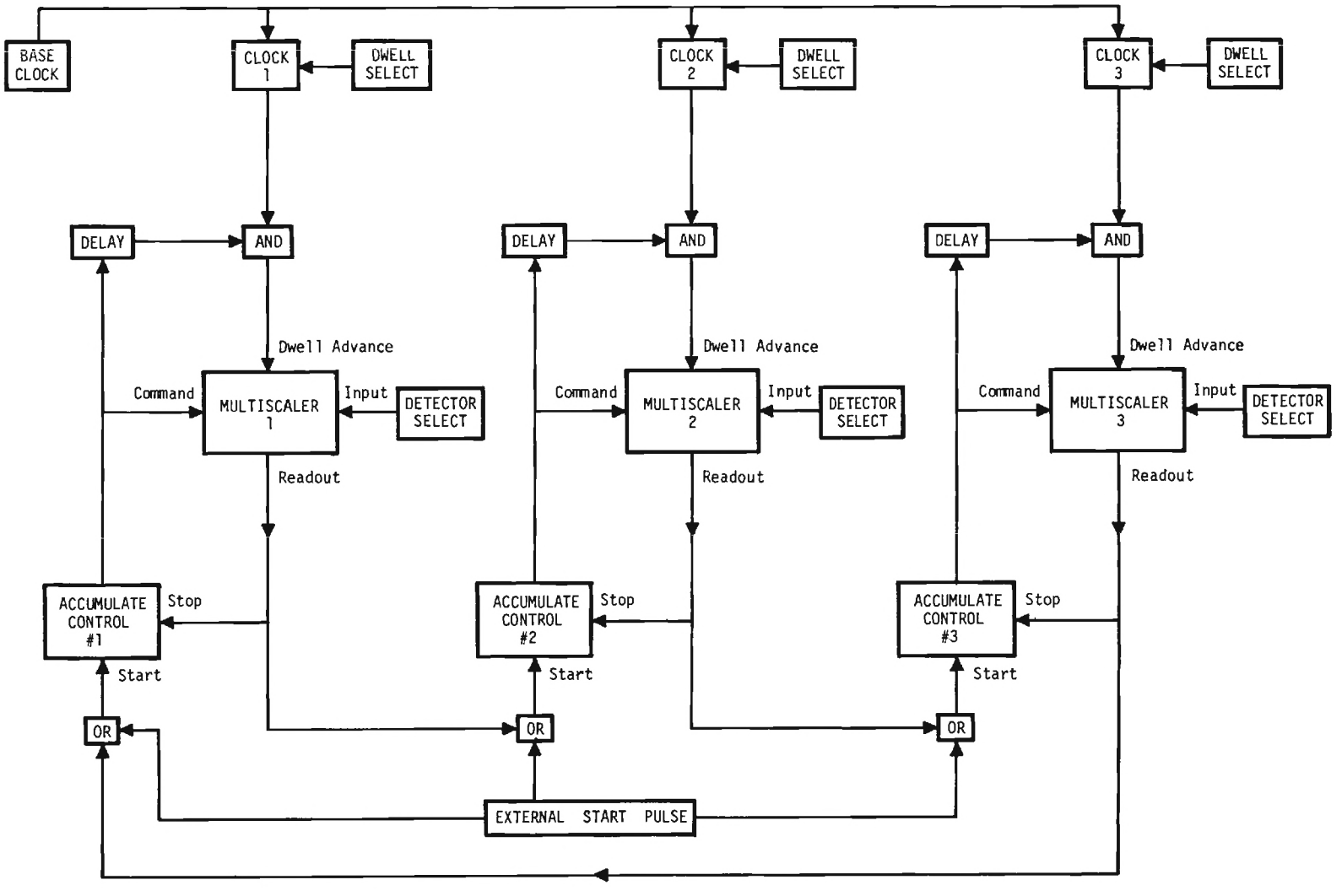


Figure 29. Schematic Block Diagram of Multiscaler Sequencing Logic

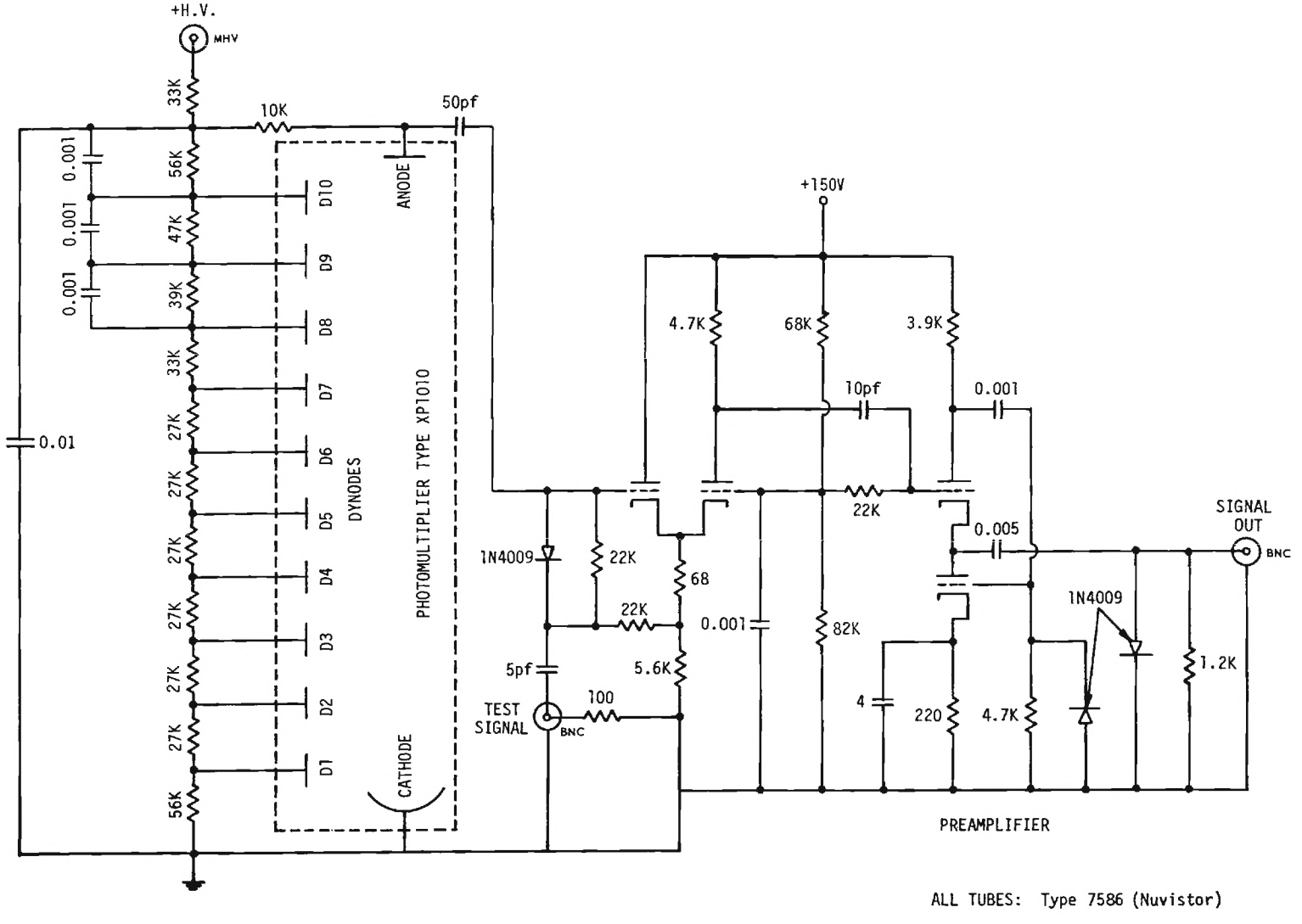


Figure 30. Schematic Diagram of Scintillation Detector High Voltage Divider and Preamplifier

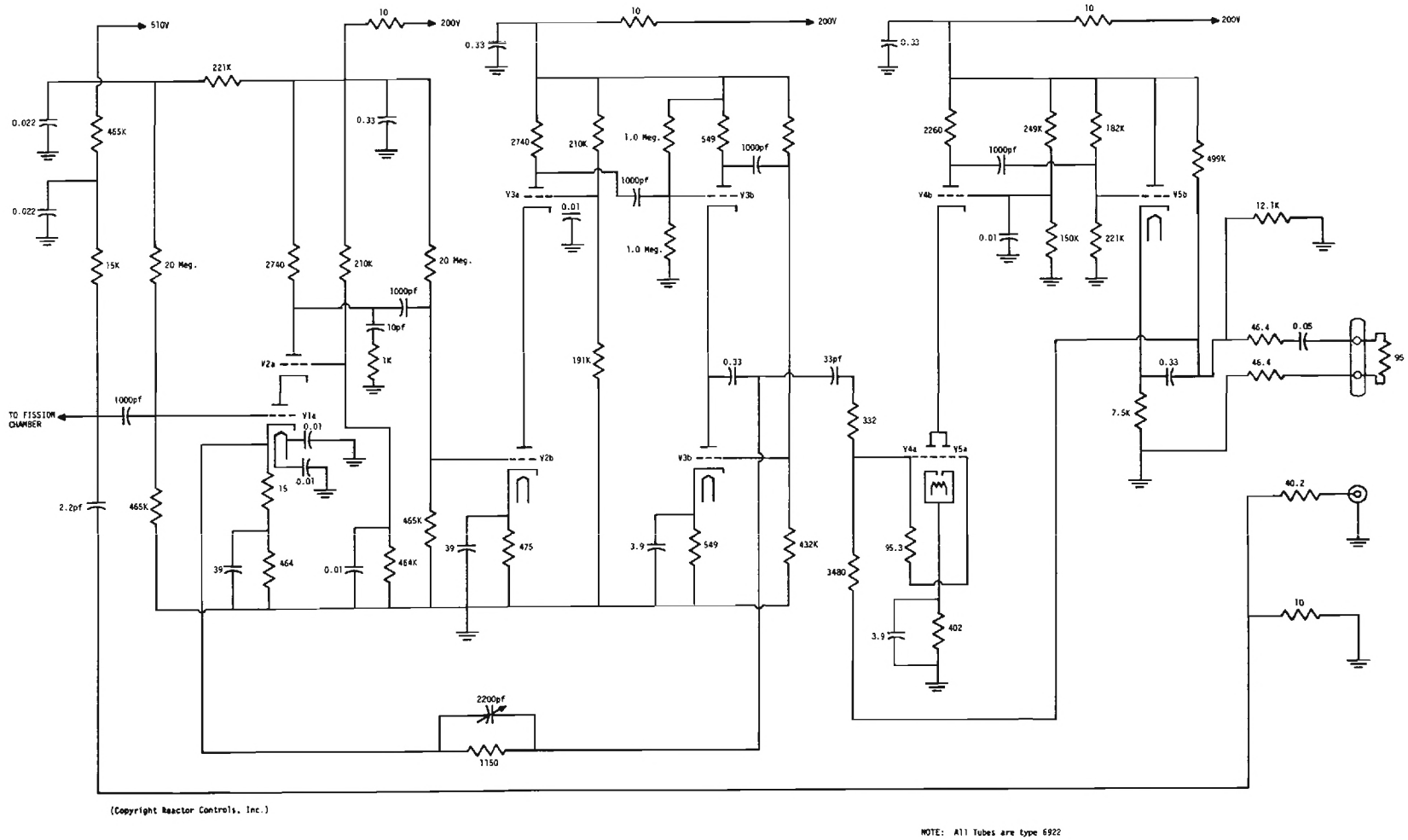


Figure 31. Schematic Diagram of Fission Chamber Preamplifier

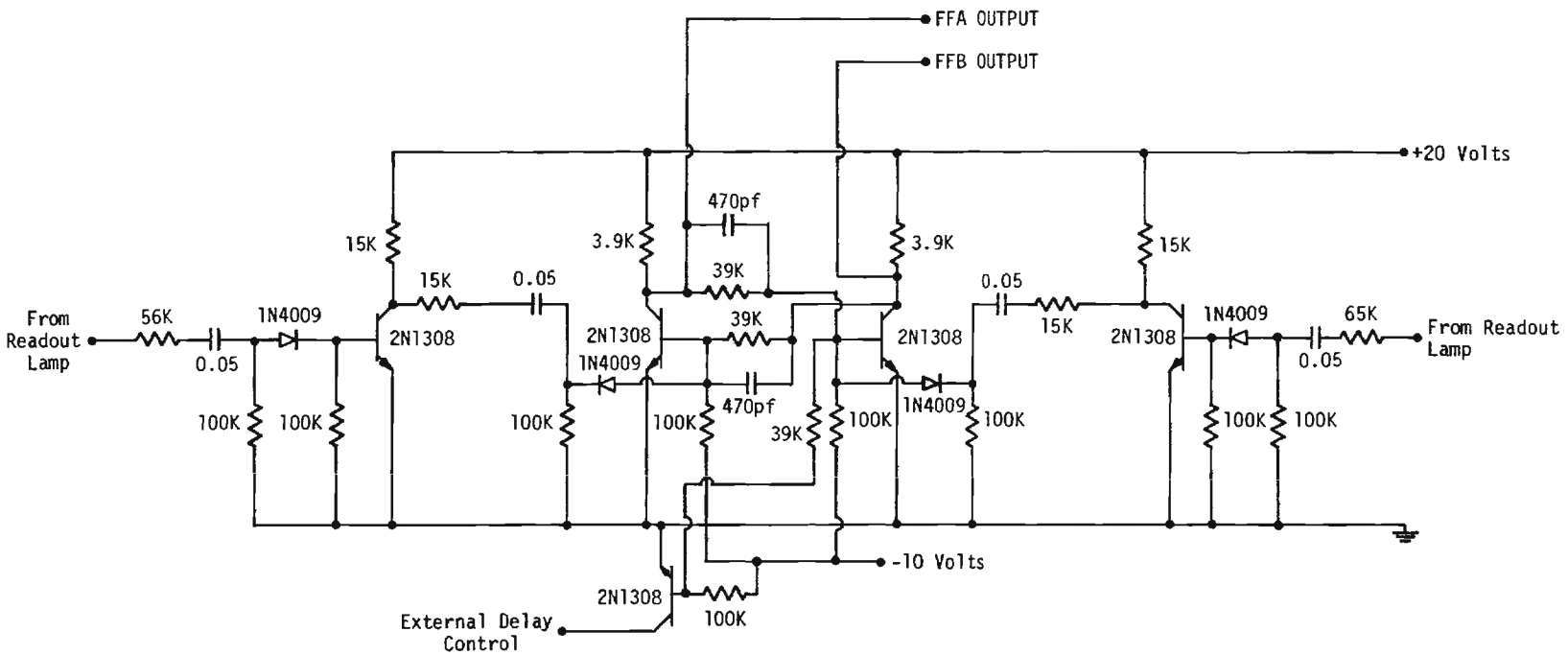


Figure 32. Schematic Diagram of Analyzer and Clock Control Circuit

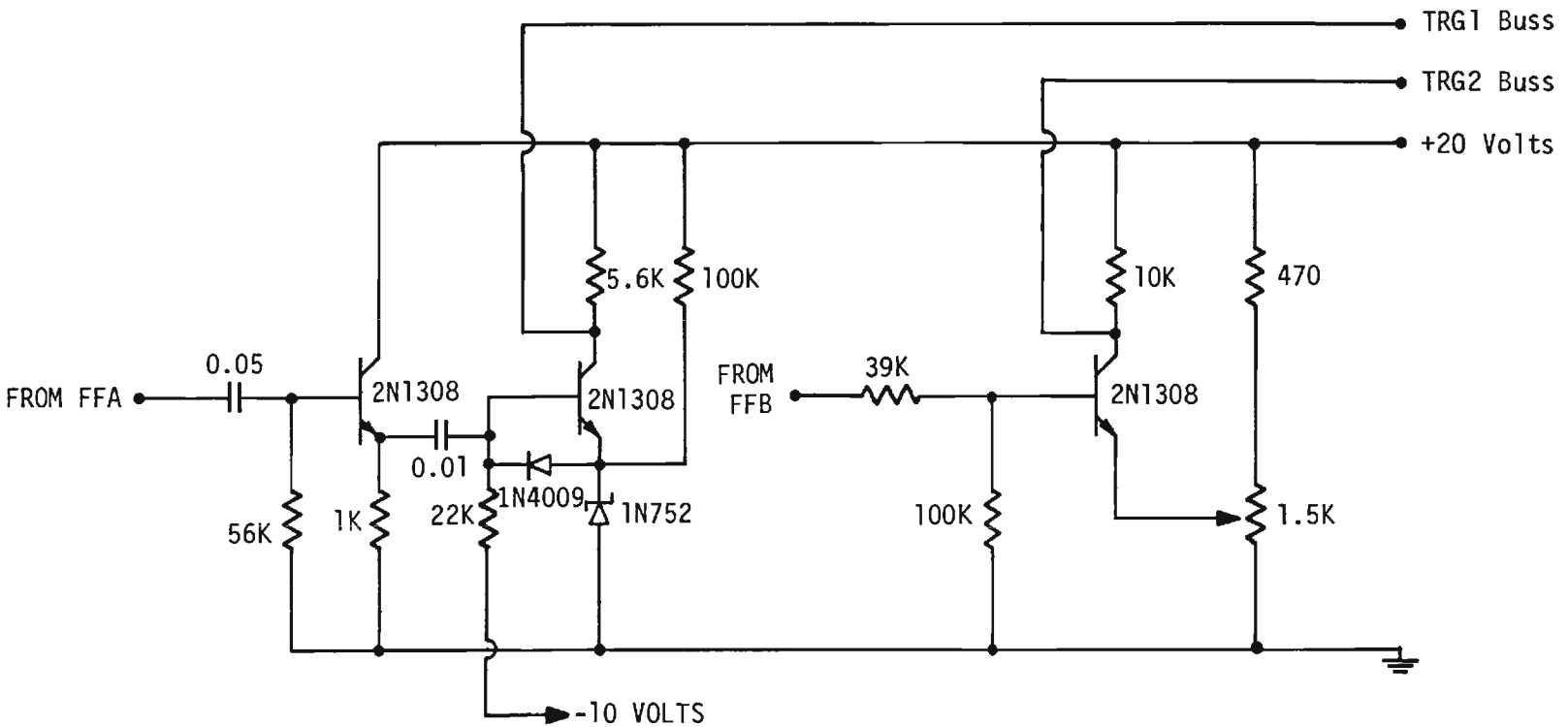


Figure 33. Schematic Diagram of Decade Divider Trigger Section

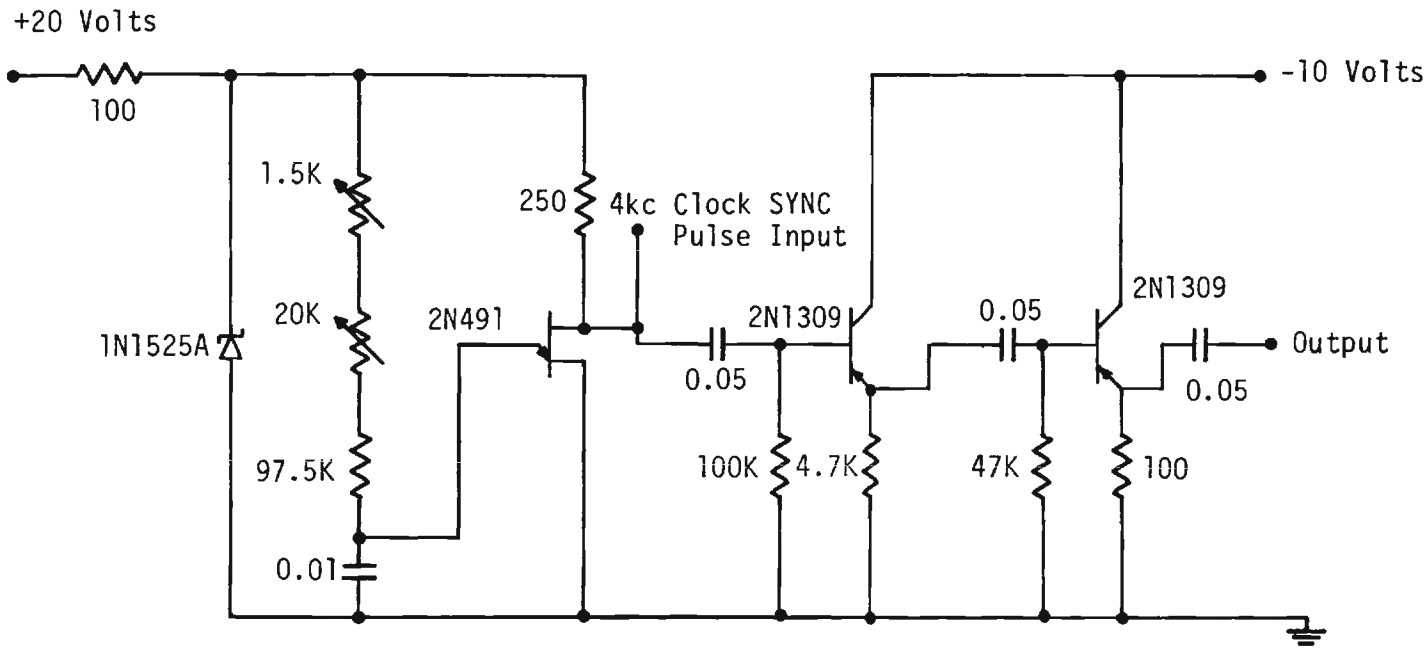


Figure 34. Schematic Diagram of 1 kc Oscillator

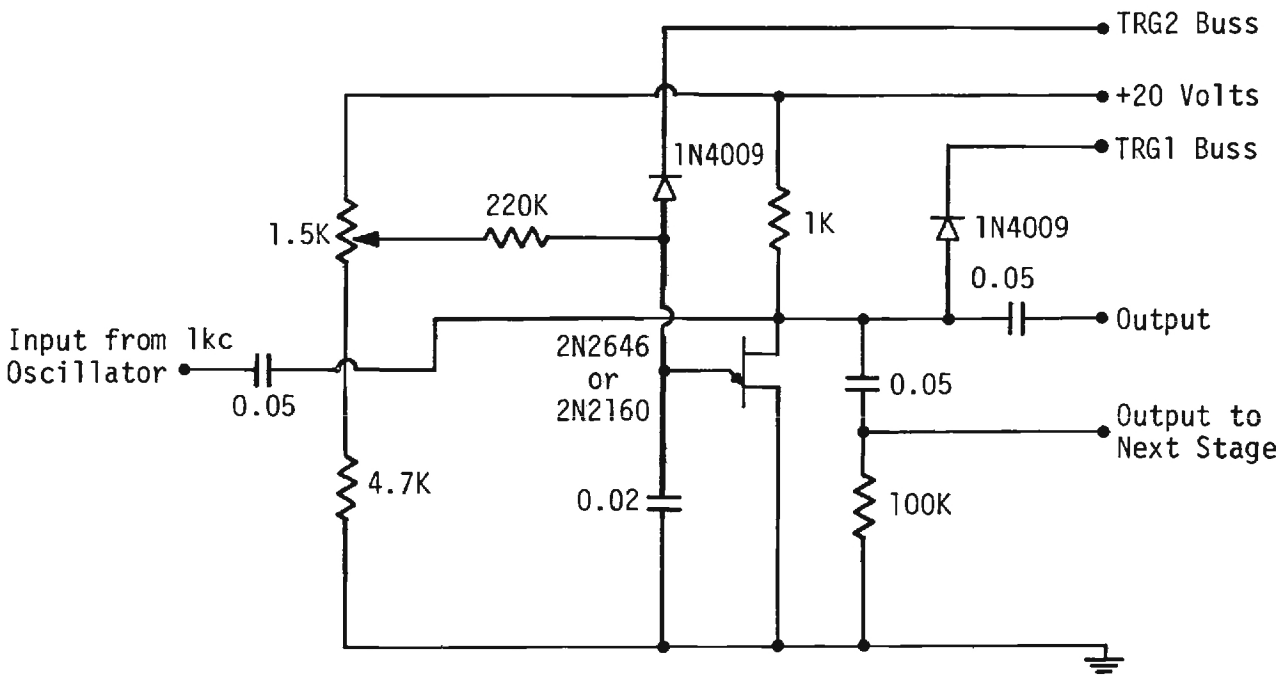
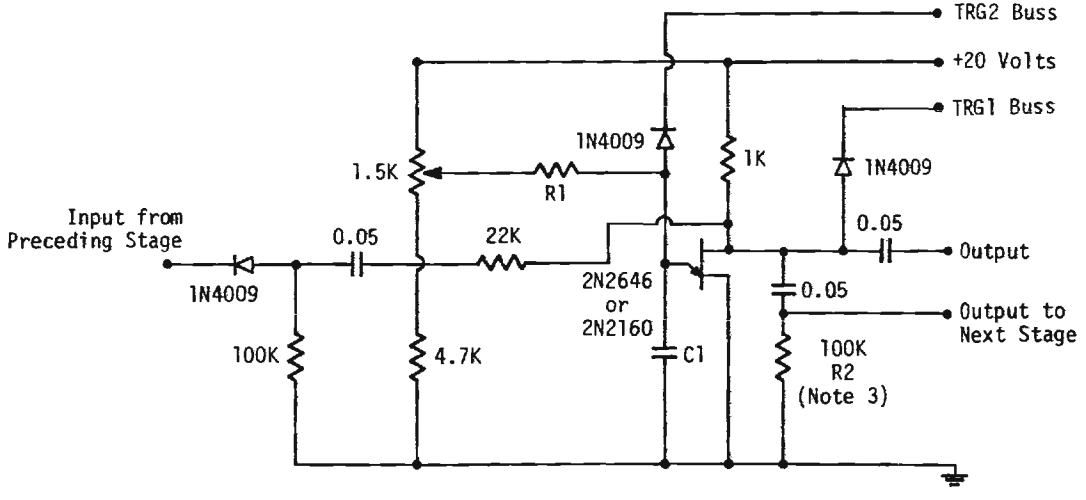


Figure 35. Schematic Diagram of Decade Divider First Section



NOTE

1. There are Four of these per Clock Unit.
2. The R1, C1, Combinations are:

STAGE	R1	C1
1	522K	0.25 µfd
2	20K	60 µfd
3	75K	60 µfd
4	950K	60 µfd

3. Resistor R2 is 150K in Stage 4.

Figure 36. Schematic Diagram of Decade Divider Intermediate Section

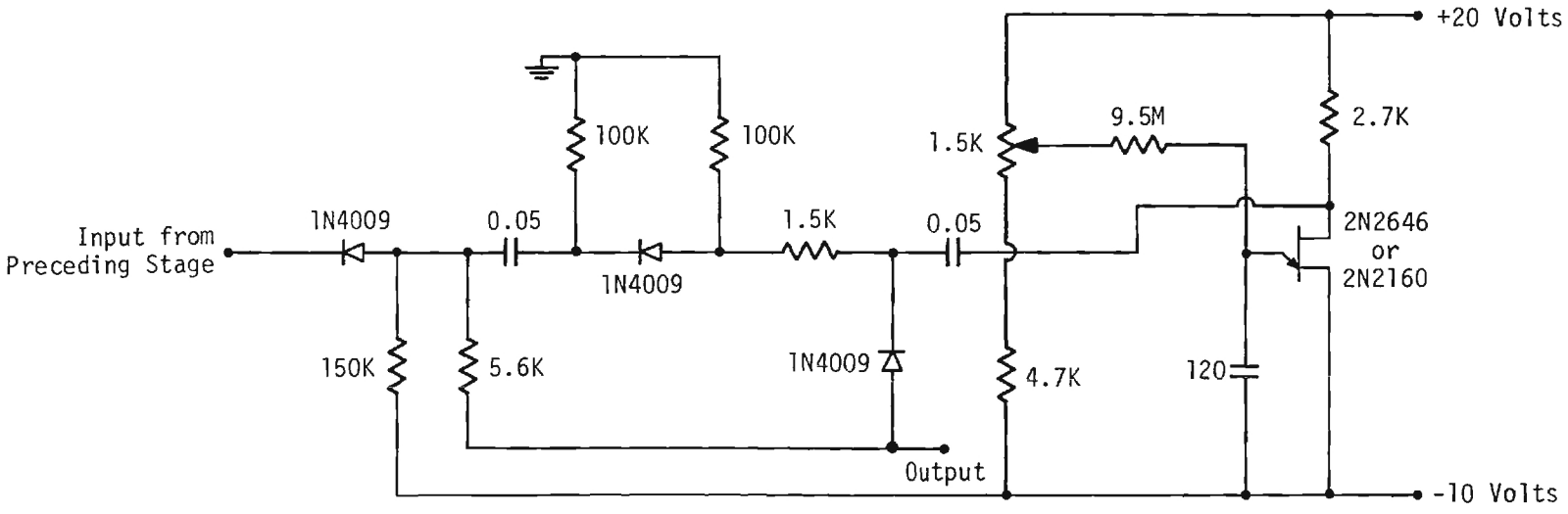


Figure 37. Schematic Diagram of Decade Divider Final Section

BIBLIOGRAPHY

1. A. M. Weinberg and E. P. Wigner, The Physical Theory of Neutron Chain Reactors (The University of Chicago Press, Chicago, 1958), p. 115.

2. R. Roberts, R. Meyer, and P. Wang, Physical Review 55, 510 (1939).

3. S. Glasstone and M. C. Edlund, The Elements of Nuclear Reactor Theory (D. Van Nostrand Co., Inc., Princeton, 1952), chap. 4.

4. A. M. Weinberg and E. P. Wigner, The Physical Theory of Neutron Chain Reactors (The University of Chicago Press, Chicago, 1958), p. 154.

5. R. C. Mobley and R. A. Laubenstein, Phys. Rev. 80, 309 (1950).

6. G. R. Keepin, Physics of Nuclear Kinetics (Addison-Wesley Publishing Co., Inc., Reading, Mass., 1965), p. 142.

7. S. Glasstone and M. C. Edlund, The Elements of Nuclear Reactor Theory (D. Van Nostrand Co., Inc., Princeton, 1952), chap. 10.

8. C. E. Cohn and B. J. Toppel, Argonne National Laboratory Report ANL-6134 (1960).

9. D. J. Hughes, J. Dabbs, A. Cahn, and D. Hall, Phys. Rev. 73, 111 (1948).

10. S. Bernstein, W. M. Preston, G. Wolfe, and R. E. Slattery, Phys. Rev. 71, 573 (1947).

11. G. R. Keepin, Nucleonics 20, No. 8, 150 (1962).

12. G. R. Keepin, T. F. Wimett, and R. K. Zeigler, Phys. Rev. 107, 1044 (1957).

13. B. P. Maksiutenko, Journal of Experimental and Theoretical Physics (USSR) 35, 815 (1958).

14. W. Spatz, D. Hughes, and A. Cahn, Phys. Rev. 72, 163 (1947).

15. O. J. Hahn and R. C. Axtmann, Princeton University Report PNS-1 (Unpublished)(1962).

16. B. S. Finn, Nuclear Science and Engineering 7, 369 (1960).

BIBLIOGRAPHY (Continued)

17. W. K. Ergen, Aircraft Nuclear Propulsion Report ANP-59 (1951).

18. R. Batchelor and H. R. McK. Hyder, Journal of Nuclear Energy 3, 7 (1956).

19. M. W. Johns and B. W. Sargent, Canadian Journal of Physics 32, 136 (1954).

20. P. R. Tunnicliffe, Atomic Energy of Canada Ltd. Report AECL-401 (1952).

21. Letter from R. W. Meier, Würenlingen, 1963.

22. G. R. Keepin, Progress in Nuclear Energy (Pergamon Press, London, 1956) Vol. I, p. 191.

23. G. R. Keepin, Physics of Nuclear Kinetics (Addison-Wesley Publishing Co., Inc., Reading, Mass., 1965) chap. 4.

24. H. S. Isbin, Nuclear Reactor Theory (Reinhold Publishing Corp., New York, 1963), p. 80.

25. R. Perez-Belles, J. D. Kington, and G. de Saussure, Nucl. Sci. and Eng. 12, 505 (1962).

26. T. J. Thompson, Private Communication, 1964.

27. "Final Safeguards Report for the Georgia Tech Research Reactor" (Unpublished)(1963).

28. R. Stedman, Review of Scientific Instruments 31, 1156 (1960).

29. U.S. Atomic Energy Commission, License No. R-97, "Technical Specifications for the Georgia Institute of Technology," Appendix A (1964).

30. C. E. Cohn and J. J. Kaganove, Transactions of the American Nuclear Society 5, 388, No. 2 (1962).

31. J. J. Kaganove, Argonne National Laboratory Report ANL-6132 (1960).

32. C. E. Cohn, Private Communication, 1964.

33. W. C. Davidon, Argonne National Laboratory Report ANL-5990 (Rev.) (1959).

34. R. Fletcher and M. J. D. Powell, The Computer Journal 7, 163 (1964).

35. R. V. Meghreblian and D. K. Holmes, Reactor Analysis (McGraw-Hill Book Company, Inc., New York, 1960), p. 614.

BIBLIOGRAPHY (Concluded)

36. S. Glasstone and M. C. Edlund, The Elements of Nuclear Reactor Theory (D. Van Nostrand Co., Inc., Princeton, 1952), p. 333.

37. W. W. Graham and D. S. Harmer, "Georgia Institute of Technology Technical Report No. NE-1" (Unpublished)(1963).

VITA

Walter Waverly Graham, III was born on April 30, 1933 in Nashville, Tennessee. He attended public schools there and in Annapolis, Maryland, and graduated from Hillsboro High School, Nashville in 1951. Between the years 1951 and 1955, Mr. Graham was a midshipman at the United States Naval Academy, serving as Brigade Commander and Editor-in-Chief of The Log Magazine. Commissioned an Ensign in 1955, he was assigned duty on USS Massey (DD778), Staff Commander Destroyer Flotilla Four, USS O'Hare (DDR889), and as a Combat Information Center Instructor at the Fleet Air Defense Training Center, Dam Neck, Virginia.

Mr. Graham left active duty in 1959 to attend graduate school at Vanderbilt University. He completed the Master of Science degree in Nuclear Engineering in 1960 and was employed by the General Atomic Division, General Dynamics Corporation, San Diego, California. At General Atomic, Mr. Graham was engaged in the design of the Peach Bottom High Temperature Gas-Cooled Reactor Fission Product Removal System. He is author of "TNT, A Program for Analyzing the Transient Performance of the Peach Bottom HTGR Fission Product Trapping System," GAMD-3168, and co-author of "The Removal of Radioactive Krypton and Xenon from a Flowing Helium Stream by Fixed-Bed Adsorption," GA-2395.

Mr. Graham is married to the former Ann Bennett of Nashville; they have two children, Kerry and Holly. He is a deacon in the Presbyterian Church, U.S. and holds the rank of Lieutenant Commander in the U.S. Naval Reserve. He is a member of the American Nuclear Society and Sigma Xi.